استعمال النظام
WINDOWS NT

تاليف: هيلين كاستر

ترجمة: مركز التعريب والبرمجة

الدار العربية للعلوم
Arab Scientific Publishers

استعمال النظام
WINDOWS NT

بسم الله الرحمن الرحيم

الطبعة الأولى

1414 هـ ـ 1993 م.

الدار العربية للعلوم

Arab Scientific Publishers

بناية الريم ـ شارع ساقية الجنزير ـ عين التينة ـ هاتف 806983-860138-811385

ص.ب. 13/5574 بيروت ـ لبنان ـ تلكس ABJAD 21583 LE

هاتف وفاكس دولي: 4782486 (212) 001

المحتويات

1

المهمّة

تدور عجلات التقدُّم ببطء في عالم أنظمة التشغيل. ويستغرق تطوير أنظمة التشغيل عدّة سنوات. وهي تبقى بعد إنجازها غير مفيدة على الإطلاق إلى أن يتم إدخال برامج تطبيقيّة إليها، تظهر قدرات أنظمة التشغيل هذه. وحتى بعد تواجد البرامج التطبيقيّة، يجب أن يتعلّم الناس طريقة إستعمالها بواسطة المستندات والتدريب والخبرة. وهذا يعني، مع الإعاقات العامّة التي تحصل خلال تطوير البرامج التطبيقيّة لأنظمة التشغيل، أن المستعمل العادي يملك ويستعمل تكنولوجيا نظام تشغيل بعمر من 10 أو 20 سنة.

وأثناء إنتظار الموافقة على أنظمة التشغيل، تتقدّم بسرعة تكنولوجيا أجهزة الحواسيب. وتصبح الحواسيب ذات المعالجات الأسرع والذاكرة الأكبر وحتى المعالجات المتعدّدة تصبح عامة الإستعمال بينما يحاول مطوّرو أنظمة التشغيل تحسين وتوسيع الأنظمة الموجودة لاستعمال المزايا الجديدة.

تُعرف الشرائح 80386 و80486 من Intel سريّة مع المعالجات المستعملة الأخرى على أنها حواسيب مجموعة التعليمات المعقّدة (أوCISC). والخاصية الرئيسيّة لها هي العدد الكبير من تعليمات الماكنة، حيث كل تعليمة واضحة وقويّة. ولقد قامت شركة Intel في السنوات الأخيرة بتطوير ماكنات متعدّدة المعالجات إعتماداً على تكنولوجيا CISC من Intel.

وفي منتصف الثمانينات، أنشأت مصانع الأجهزة نوع آخر من المعالجات سُمّي حواسيب مجموعة التعليمات المخفّضة (أوRISC). تختلف شرائح RISC عن شرائح CISC بشكل أساسي في العدد الصغير من تعليمات الماكنة السهلة التي توفّرها شريحة RISC. ونتيجة لسهولة مجموعة تعليماتها، تعمل معالجات RISC بسرعة وتحقّق أوقات تنفيذ سريعة جداً.

لقد ظهرت تكنولوجيا معالجات واعدة في مجالَي شرائح CISC وRISC. ولقد إغتنمت شركة Microsoft ذلك لكي تستغلّ ميزات هذه الأجهزة الأخرى التي تحتاجها لإنتاج نظام تشغيل للتسعينات وهو نظام نقّال ويستطيع التنقُّل بسهولة من منصّة جهاز إلى آخر. ورغم أن

شركتَي Microsoft و IBM أنشأتا نظام التشغيل OS/2 في الثمانينات، فقد إنتبهت شركة Microsoft إلى وجود عدّة نواقص في النظام، وأكثرها أن النظام OS/2 ليس نقّالاً. وقد كُتب بلغة Assembly ليعمل على معالج واحد، حواسيب 80286 من Intel. وعِوَضاً عن تجديد براجيّات النظام OS/2، قرّرت شركة Microsoft إنشاء نظام تشغيل جديد نقّال.

1-1 نظام تشغيل للتسعينات:

في خريف العام 1988، استخدمت شركة Microsoft السيّد David n. Cutler («Dave») لقيادة جهود تطوير براجيات جديدة، لإنشاء نظام تشغيل لشركة Microsoft للتسعينات. وجمع Dave، وهو مهندس معروف لأنظمة الحواسيب الصغيرة، فريق مهندسين لتصميم نظام التشغيل New Technology (NT) أو التكنولوجيا الحديثة. وفي بداية العام 1989، إجتمع Bill Gates ومخطّطو شركة Microsoft الأساسيّين لمراجعة مواصفات نظام التشغيل التي حدّدها فريق Dave Cutler. فمخطّطاتهم حدّدت متطلّبات السوق الأساسيّة لنظام التشغيل الحديث.

النقليّة: تتطوّر الأجهزة بسرعة وغالباً دون توقّع. فالمعالجات RISC تبتعد إلى حدٍّ كبير عن ثقافة CISC التقليديّة. ان كتابة NT في لغة نقّالة تتيح لها التنقّل بحريّة من معالج إلى آخر.

المعالجة المتعدّدة وقابليّة تدريج القياس: يجب أن تتمكّن البرامج التطبيقيّة من إستعمال ميزات المجال الواسع من الحواسيب المتوفّرة حالياً. فمثلاً، يعرض في السوق حواسيب ذات عدّة معالجات لكن يوجد عدد قليل جداً من أنظمة التشغيل التي تستخدمها. وعن طريق جعل NT نظام تشغيل متعدّد المعالجة ذات قابليّة لتدريج القياس سيتيح للمستعمِل تشغيل نفس التطبيق على حواسيب أحاديّة المعالجة والمتعدّدة المعالجات. وإحدى مزاياه المتقدّمة هي إمكانيّة المستعمل تشغيل عدّة برامج تطبيقيّة في نفس الوقت عند السرعة الكاملة، وتستطيع البرامج التطبيقيّة المكثّفة الحاسوبيّة توفير أداءاً محسّناً عن طريق تقسيم عملها على عدّة معالجات.

الحوسبة الموزّعة: مع تزايد توفّر الحواسيب الشخصيّة في الثمانينات، فقد تمّ تعديل الطبيعة الفعليّة للحوسبة نهائياً. وحين كان يخدم حاسوب إيواني كبير واحد شركة بأكملها، تكاثرت الحواسيب الصغيرة الأصغر والأرخص وهي تُستعمل حالياً من قِبَل جميع الموظفين. وتتيح قدرات شبكة الإتصال للحواسيب الأصغر الإتصال مع بعضها البعض ومشاركة موارد الأجهزة مثل فسحة القرص أو طاقة المعالجة (على شكل ملقّمات ملفّات أو ملقّمات طباعة أو ملقّمات إحتساب). ولاستيعاب هذا التغيير، يبني مصمّمو النظام NT قدرات شبكة إتصال

مباشرة في نظام التشغيل ويوفروا الوسائل للبرامـج التطبيقية لتوزيـع عملها على أنظمة الحاسوب المتعدّدة.

التوافق مع النظام POSIX: من منتصف إلى أواخر الثمانينات، بدأت الوكالات الحكوميّة الأميركيّة بتحديد POSIX كمقياس شرائي لعقود شراء الحواسيب الحكوميّة. والنظام POSIX هو نظام «تداخل نظام التشغيل النقّال وفقاً للنظام UNIX» وينسب إلى مجموعة من المعايير الدوليّة لأنظمة تداخل نظام التشغيل من النوع UNIX. يشجّع نظام POSIX (مواصفات IEEE رقم 1003.1 - 1988) البائعين على إستخدام أنظمة التداخل من النوع UNIX لجعلهم متوافقين، ولكي يتمكّن المبرمجون من نقل برامجهم التطبيقيّة بسهولة من نظام إلى آخر. ولموافقة المتطلّبات الشرائيّة POSIX الحكوميّة، يضمّ النظام NT ليوفّر محيط تشغيل برنامـج تطبيقي POSIX خياري.

نظام الأمان المرخّص من الحكومة الأميركيّة: إضافة إلى التوافق مع النظام POSIX، تحدِّد الحكومة الأميركيّة أيضاً خطوط توجيه لنظام أمان الحواسيب لبرامج الحكومة التطبيقيّة. إن الحصول على تصنيف نظام الأمان المرخّص من الحكومة يتيح المنافسة لنظام التشغيل في هذا المجال. وهذه القدرات المطلوبة هي طبعاً مزايا متعدّدة لأي نظام متعدِّد الإستعمال. تحدّد خطوط توجيه الأمان القدرات المطلوبة مثل حماية موارد مستعمل واحد من موارد مستعمل آخر ووضع حصّة نسبيّة للموارد لمنع مستعمل واحد من تجميع كل موارد النظام (كالذاكرة مثلاً).

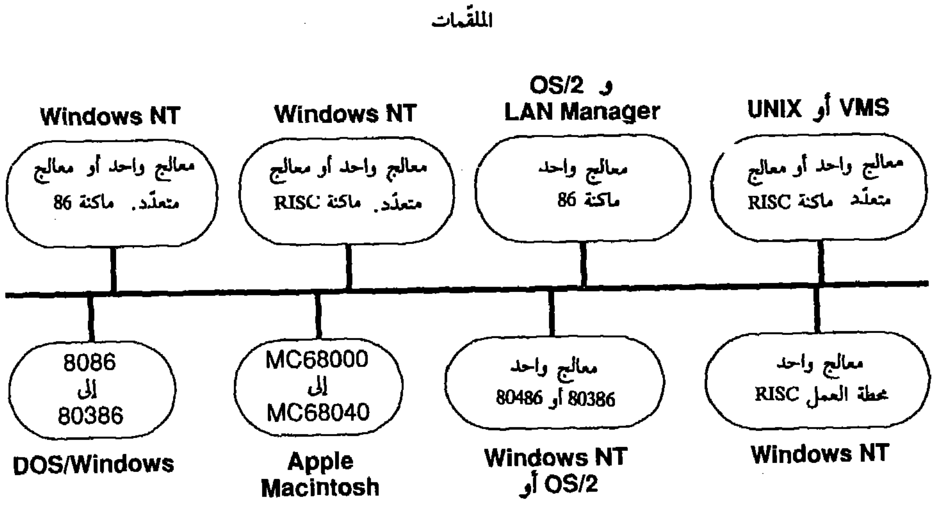
هذا الهدف الرئيسي لنظام الأمان في NT يسمّى المستوى Class C2 والمُعرَّف من قِبَل وزارة الدفاع الأميركيّة كتوفير «حماية سرّية (على أساس الحاجة للمعرفة) وعبر شمل قدرات التدقيق في مسؤوليّة الموظفين والأعمال التي حفّزوها». وهذا يعني أن مالك موارد النظام يستطيع تحديد من يستطيع الوصول إليها، وأن نظام التشغيل يستطيع إكتشافه عندما يتمّ الوصول إلى البيانات ومن قِبَل من. تمتدُّ مستويات الأمان الحكوميّة الأميركيّة من المستوى D (الأقل تشديداً) إلى المستوى A (الأكثر تشديداً) مع إحتواء المستويين B و C على مستويات فرعيّة. ورغم أن NT سيكتب مبدئياً ليدعم مستوى الأمان C2، قد تلائم الإصدارات المستقبليّة المتطلّبات الأكثر تشديداً لمستويات الأمان الأعلى.

وبوضع متطلّبات السوق هذه قيد التنفيذ، حقّق فريق تطوير NT مهمّته في إنشاء نظام تشغيل للتسعينات من قبَل Microsoft. في البدء كانت الخطّة في جعل NT يحتوي على نظام تداخل مع المستعمل من النوع OS/2 وتوفير نظام تداخل برمجة البرنامج التطبيقي OS/2 (API) كنظام تداخل البرمجة الأولي. وخلال تطوير النظام، نزل إلى السوق الإصدار 3,0 لبرنامج Windows من Microsoft وحقّق نجاحاً باهراً بعكس النظام OS/2.

نتيجة لهذا الواقع والتعقيدات المشمولة في تحسين ودعم نظامَي تشغيل غير متوافقين، قرّرت شركة Microsoft تعديل مسارها وتوجيه طاقاتها بإتجاه إستراتيجيّة نظام تشغيل متماسك واحد. فالإستراتيجيّة هي في إنتاج مجموعة من أنظمة تشغيل تعتمد على النظام Windows والتي تغطي حواسيب المفكرات الصغيرة إلى أكبر محطات عمل المعالجات المتعدّدة. إن النظام NT Windows، كما سمّي الجيل التالي من نظام Windows، ينفذ عند المزيّة المتقدّمة لنظام التشغيل المعتمد على Windows لإمداد النظام Win 32 API نظام تداخل البرمجة من 32 بت لتطوير برنامج جديد. ويوفّر النظام WIN 32 API قدرات نظام التشغيل المتقدّم إلى التطبيقات عبر العديد من المزايا مثل المعالجات المتعدّدة الشعب والمزامنة والأمان ونظام الدخل / الخرج وإدارة الكائنات.

إن النظام Windows NT غير موجود في فراغ. فهو يستطيع شمل أنظمة Microsoft الأخرى مع Apple Macintosh ومع أنظمة التشغيل المتوافقة مع UNIX على Microsoft LAN Manager أو شبكة أخرى. يظهر مثال على تشكيل النظام في الشكل (1-1).

تستطيع الملقّمات في هذا التشكيل توفير البرامج الخدماتيّة لنظام التشغيل مثل ملقّمات الملفّات وملقّمات الطباعة أو وظائف إدارة النظام أو تستطيع توفير برامج خدماتيّة للبرنامج مثل ملقّمات قاعدة البيانات Database. وقد يتفاعل برنامج تطبيقي مع الملقّم دون علم المستعمل.



الشكل (1-1)
توصيل عدّة أنظمة

وعند تشكيله كملقّم، يعمل النظام Windows NT كنظام تشغيل لعدّة مستعملين، حيث يخدم حاجات عدد كبير من المستعملين على شبكة. ويمكن لكل محطة عمل دعم مستعمل متفاعل واحد وعدّة مستعملين بعيدين، مع ضرورة تسجيل كل مستعمل (أو برنامج) قبل الوصول إلى النظام.

2-1 الأهداف التصميميّة:

يحتاج تصميم براجيّات النظام Windows NT لبعض التفكير العميق. ولكي يفي النظام بمتطلّبات السوق، كان من المفروض شمل المزايا المعقّدة مثل التوافق مع POSIX والأمان من البداية.

وقبل البدء بكتابة عدّة مئات الآلاف من أسطر الشيفرة التي سيتألّف منها النظام Windows NT، حدّد مهندسو النظام مجموعة من الأهداف التصميميّة للبراجيّات. وقد سهّلت هذه الأهداف التصميميّة إتخاذ الآف القرارات الإضافيّة التي حدّدت البنية الداخليّة لمشروع براجيّات كبير. وعندما يتعارض خياران تصميميّان، تساعد الأهداف التصميميّة على إختيار الأفضل. إن ما يلي هي الأهداف التصميميّة للنظام Windows NT:

- **المدوديّة**: يجب أن تُكتب الشيفرة لتكبر بشكل مريح وتتغيّر وفقاً لتغيُّر متطلّبات السوق.
- **النّقليّة**: كما حُدّدت من قبل أهداف السوق، يجب أن تتمكّن الشيفرة من التحرُّك من معالج واحد إلى آخر.
- **الإعتماديّة والقوّة**: يجب أن يحمي النظام نفسه من الأعطال الداخليّة ومحاولات العبث الخارجيّة. ويجب أن يعمل كما هو متوقّع في كل الأوقات، ولا يجب أن تتمكّن البرامج التطبيقيّة من إلحاق الأذى بنظام التشغيل أو بوظيفته.
- **التوافقيّة**: رغم أنه لا ضرورة لكي يقوم النظام Windows NT بتوسيع الثقافة الموجودة، فإن نظامَي التداخل مع المستعمل و API يجب أن يتوافقا مع أنظمة Microsoft.
- **الأداء**: ضمن تقييدات الأهداف التصميميّة الأخرى، يجب أن يكون النظام سريعاً وإستجابياً قدر الإمكان على كل منصّة جهاز.

تشرح الأقسام التالية الأهداف التصميميّة للنظام Windows NT بتفصيل أكبر وتصف تأثيرها على النسق الأخير لنظام التشغيل.

1-2-1 المدوديّة:

تتغيّر أنظمة التشغيل على مرّ الزمن. وتتمثّل هذه التغييرات عادة بشكل تصاعدي على شكل مزايا جديدة: فمثلاً، دعم جهاز جديد مثل قارىء الأقراص CD-ROM وقدرة الإتصال مع نوع شبكة جديد، أو دعم تقنيّات براجميات ستصدر، مثل أنظمة التداخل التخطيطيّة مع المستعمل أو محيط البرمجة الكائنيّة.

لقد كان الهدف التصميمي الأول هو ضمان تكامل النظام Windows NT عندما يتغيّر نظام التشغيل مع الوقت. وبالنسبة لنظام التشغيل Mach الذي طُوِّر في جامعة Carnegie-Mellon، إعتمد الدكتور Richard Rashid وزملاؤه طريقة فريدة لهذه المشكلة عن طريق إنشاء قاعدة نظام تشغيل توفّر القدرات المبدئيّة لنظام التشغيل. وتوفّر البرامج التطبيقيّة التي تسمّى ملقّمات قدرات إضافيّة لنظام التشغيل، بما فيها نظام API كامل المزايا. ويبقى قسم القاعدة من النظام مستقراً بينما يتمُّ تحسين الملقّمات أو يتمُّ إنشاء ملقّمات جديدة وفقاً لتغيُّر المتطلّبات.

إستعار النظام Windows NT هذا التصميم ويتألف من ملقّم تنفيذي بأفضليّة، وملقّمات بأفضليّة أدنى تسمّى الأنظمة الفرعيّة المحميّة. وينسب التعبير أفضليّة إلى أنماط تشغيل المعالج. تتّصف معظم المعالجات بنمط أفضليّة (أو عدّة أنماط) حيث تُتاح كل تعليمات الماكنة ويُتاح الوصول إلى ذاكرة النظام، والنمط بأفضليّة أدنى حيث لا يسمح بتعليمات معيّنة ولا يُتاح الوصول إلى ذاكرة النظام. وفي مصطلحات النظام Windows NT، يسمى نمط المعالج بأفضليّة نمط النواة ويسمّى نمط المعالج بأفضليّة أدنى نمط المستعمل.

يشتغل عادة نظام تشغيل في نمط النواة وتشتغل البرامج التطبيقيّة فقط في نمط المستعمل باستثناء عند تحديدها ملقّمات لنظام التشغيل.

إن تصميم النظام Windows NT فريد من نوعه لأن الأنظمة الفرعيّة المحميّة تعمل في نمط المستعمل كما تعمل البرامج التطبيقيّة. وتتيح هذه البنية تعديل الأنظمة الفرعيّة المحميّة أو إضافتها دون التأثير على تكامل الملقّم التنفيذي. (راجع الفصل الخامس «Windows والأنظمة الفرعيّة المحميّة»).

إضافة إلى الأنظمة الفرعيّة المحميّة، يتضمّن النظام Windows NT عدّة مزايا أخرى لضمان مدوديّته:

■ بنية منظوميّة: يحتوي الملقّم التنفيذي مجموعة من المكوّنات الإفراديّة التي تتفاعل مع بعضها البعض فقط عبر أنظمة التداخل الوظيفيّة. ويمكن إضافة مكوّنات جديدة إلى الملقّم التنفيذي

بطريقة منظوميّة وتنفيذ عملها بواسطة إستدعاء أنظمة التداخل المزوّدة من قِبَل المكوّنات الموجودة.

- **إستعمال الكائنات لعرض موارد النظام**: تتيح الكائنات، وهي أنواع بيانات تجريديّة تعالَج فقط من قِبَل مجموعة خاصّة من ملقّمات الكائنات، معالجة موارد النظام بشكل متناسق. ولا يؤثر إضافة كائنات جديدة على الكائنات الموجودة أو يتطلّب تغيير الشيفرة الموجودة. (راجع الفصل الثالث «برنامج إدارة الكائنات وأمان الكائنات» للحصول على مزيد من المعلومات).

- **السوّاقات المحمّلة**: يدعم نظام الدخل / الخرج في النظام Windows NT السوّاقات التي يمكن إضافتها إلى النظام خلال تشغيله. ويمكن دعم أنظمة ملفّات جديدة وأجهزة وشبكات عن طريق كتابة مسيّق جهاز ومسيّق نظام الملفّات أو مسيّق النقل وتحميلها في النظام. (راجع الفصل الثامن «نظام الدخل / الخرج» والفصل التاسع «إنشاء الشبكات» للحصول على مزيد من المعلومات).

- **برنامج إستدعاء إجراء بعيد (RPC) الخدماتي**: الذي يتيح لبرنامج تطبيقي إستدعاء ملقّمات بعيدة دون إعتبار لموقعها على الشبكة. ويمكن إضافة ملقّمات جديدة إلى أي ماكنة على الشبكة وجعلها متوفّرة فوراً للبرامج التطبيقيّة على الماكنات الأخرى على الشبكة (راجع الفصل التاسع «إنشاء الشبكات» للحصول على مزيد من المعلومات).

2-2-1 النّقليّة:

يتعلّق الهدف التصميمي الثاني، التوافقيّة بين الأنظمة المختلفة، بالمدوديّة. تتيح المدوديّة تحسين نظام تشغيل بسهولة بينما تمكّن التوافقيّة بين الأنظمة المختلفة نقل نظام تشغيل بأكمله إلى ماكنة تعتمد على معالج أو تشكيل مختلف، مع أدنى قدر ممكن من التسجيل. ورغم وصف أنظمة التشغيل على أنها إمّا «نقّالة» أو «غير نقّالة»، فإن النّقليّة ليست حالة ثنائيّة لكنّها عبارة عن درجة. لكن السؤال المهمّ هو ليس إمكانيّة نفاذ البراجيّات (معظمها سينفذ) بل مدى صعوبة نفاذها.

إنّ كتابة نظام تشغيل سهل النفاذ مشابه لكتابة أي شيفرة نقّالة ــ يجب إتّباع خطوط توجيه معيّنة. أولاً، كتابة قدر ما يمكن من الشيفرات في لغة متوفّرة على كل الماكنات حيث تريد النفاذ. وهذا يعني عادة أنه يجب كتابة الشيفرة في لغة عالية المستوى، ويفضّل لغة تمّ استعمالها قياسياً. إن لغة التجميع assembly ليست نقّالة متأصّلة ما لم تردْ النفاذ إلى ماكنات ذات تعليمات متوافقة مع الماكنة (مثل التنقّل من Intel 80386 إلى Intel 80486 على سبيل المثال).

ثانياً، يجب إعتبار المحيط الفعلي إلى حيث تريد أن تنفذ البراجيات. تفرض الأجهزة المختلفة تقييدات مختلفة على نظام تشغيل. فمثلاً، لا يستطيع نظام تشغيل مبني على عناوين من 32 بتاً (إلا بصعوبة بالغة) النفاذ إلى ماكنة بعناوين من 16 بتاً.

ثالثاً، من الضروري تخفيض كميّة الشيفرات التي تتفاعل مع الجهاز أو إزالتها حيث أمكن. تتخذ تبعيّة الجهاز أشكال مختلفة. وتتضمّن بعض التبعيّات الواضحة مسجلات معالجة مباشرة وبنيات الأجهزة الأخرى أو إفتراض تشكيل أو قدرة جهاز معيّن.

رابعاً، عند تعذّر تجنّب شيفرة تعتمد على الجهاز، فإنه يجب عزلها إلى وحدات قليلة سهلة العثور. ولا يجب نشر الشيفرة التي تعتمد على الجهاز في نظام التشغيل. يعمل خطّا التوجيه الآخران جنباً إلى جنب. فمثلاً، يمكن إخفاء بنية تعتمد على الجهاز ضمن نوع البيانات عوَضاً عن الجهاز باستعمال مجموعة من الروتينات الوراثيّة. وعند إنفاذ نظام التشغيل، يجب تغيير فقط نوع البيانات والروتينات الوراثيّة.

صمّم النظام Windows NT بشكل أولي في اللغة C، مع تمديدات للنظام Windows NT المبني بإستثناء مناولة الناحية المعماريّة. ولقد إختيرت اللغة C لأنها شائعة الاستعمال ولأن مصرّفات C وأدوات تطوير البراجيّات متوفرة كثيراً في الأسواق. إضافة إلى اللغة C، تمّ كتابة أجزاء صغيرة من النظام باللغة ++C، بما فيها مكوّنات الرسوم التخطيطيّة لمحيط Windows وأجزاء من نظام تداخل المستعمل في الشبكة. واستعملت اللغة assembly فقط لأجزاء من النظام حيث يجب الإتصال مباشرة مع الجهاز (مناول المصيدة مثلاً) وللمكوّنات التي تتطلّب سرعة مثلى (مثل العمليّات الحسابيّة الصحيحة الدقيقة). ولكن، تمّ عزل الشيفرة غير النقّالة ضمن المكوّنات التي تستعملها.

- **عزل المعالج**: يجب أن تتمكّن أجزاء معيّنة بمستوى منخفض من نظام التشغيل الوصول إلى بنيات البيانات التي تعتمد على المعالج ومسجلاتها. لكن الشيفرة التي تقوم بذلك موجودة في وحدات صغيرة يمكن إستبدالها بوحدات نظيريّة للمعالجات الأخرى

- **عزل المنصّة**: يغلّف النظام Windows NT الشيفرة التي تعتمد على المنصّة داخل مكتبة الربط الديناميكي المعروفة بإسم (طبقة تجريد الجهاز) (HAL). إن تبعيّات أو إعتماديّة الجهاز هي تلك التي تتغيّر بين محطّتي عمل البائعين المبنيّة حول نفس المعالج ــ مثلاً، MIPS R4000. تجرّد HAL الجهاز، مثل المخابىء ووحدات التحكّم بمقاطعة الدخل / الخرج، بطبقة من البرمجيّات ذات المستوى المنخفض بحيث لا حاجة لتغيير الشيفرة ذات المستوى الأعلى عند التنقّل من منصّة إلى أخرى.

لقد كُتب النظام Windows NT لتسهيل النفاذ إلى الماكنات التي تستعمل عناوين خطيّة من 32 بتاً وتوفير قدرات ذاكرة ظاهريّة. ويستطيع الإنتقال إلى ماكنات أخرى لكن عند كلفة أكبر.

3-2-1 الإعتماديّة:

الإعتماديّة كانت الهدف التصميمي الثالث لشيفرة Windows NT. تنسب الإعتماديّة إلى فكرتين مختلفتين لكن متعلقتين. الأولى، هي أنه يجب على نظام التشغيل أن يكون قوياً، بحيث يستجيب كما هو متوقّع لحالات الخطأ وحتى التي تنتج عن تعطّل الجهاز. الثانية، هي أنه يجب على نظام التشغيل أن يحمي بشكل فاعل نفسه ومستعمليه من الأضرار العرضيّة أو المقصودة نتيجة برامج المستعمل.

المناولة الإستثنائيّة البنيويّة هي طريقة لإلتقاط حالات الخطأ والإستجابة لها بشكل متناسق. وهي عامل دفاع أولي في النظام Windows NT ضد الأخطاء في البراعجيات أو الأجهزة. ويصدر إما نظام التشغيل أو المعالج إستثناءاً عند حصول حدث غير عادي. وتحفز شيفرة المناولة الإستثنائيّة، المتواجدة في النظام، تلقائياً إستجابةً للحالة بحيث لا يلحق أي خطأ غير مكتشف الأذى ببرامج المستعمل أو بالنظام نفسه. (راجع الفصل الثاني، «نظرة شاملة حول النظام»، للحصول على مزيد من المعلومات).

تعزّز القوّة من قِبَل المزايا الأخرى لنظام التشغيل:

- تصميم منظومي يقسّم الملقّم التنفيذي إلى سلسلة من الرزمات المرتّبة. تتفاعل مكوّنات النظام الإفراديّة مع بعضها البعض عبر أنظمة تداخل البرمجة المحددة بعناية فيمكن، على سبيل المثال، إزالة مكوّن كبرنامج إدارة الذاكرة، كاملاً وإستبداله ببرنامج إدارة ذاكرة جديد يستخدم نفس أنظمة التداخل. (راجع الفصل الثاني «نظرة شاملة حول النظام»، للحصول على مزيد من المعلومات).
- نظام ملفّ جديد مصمّم للنظام Windows NT، يسمّى نظام ملفّات NT (NTFS). يستطيع نظام الملفّات NTFS إستعادة كل أنواع أخطاء القرص بما فيها الأخطاء التي تنتج في قطاعات القرص الحرجة. وهو يستعمل تخزين فائض ومخطّط تعاملي لتخزين البيانات لضمان الإستعادة.

إن المزايا التالية لنظام Windows NT تحميه من الإعتداءات الخارجيّة:

- تصميم أمان مرخّص من الحكومة الأميركيّة يوفّر مجموعة من آليّات الأمان، مثل تسجيل

المستعمل وحصص الموارد ووقاية الكائن. (راجع الفصل الخامس «Windows والأنظمة الفرعيّة المحميّة»، للحصول على مزيد من المعلومات).

- الذاكرة الظاهريّة، التي تزوّد كل برنامج بمجموعة كبيرة من العناوين التي يمكن أن يستعملها. وعندما يصل البرنامج إلى هذه العناوين الظاهريّة، فإن برنامج إدارة الذاكرة يخطّطها أو يترجمها إلى مواقع ذاكرة فعليّة. ولأنه يتحكّم بوضعيّة كل برنامج في الذاكرة، يمنع نظام التشغيل مستعملاً واحداً من قراءة أو تعديل الذاكرة التي يشغلها مستعمل آخر إلا إذا تشارك المستعملون الذاكرة بشكل واضح. (راجع الفصل السادس «برنامج إدارة الذاكرة الظاهريّة» للحصول على مزيد من المعلومات).

4-2-1 التوافقيّة:

إن توافقيّة البراجيّات، الهدف التصميمي الرابع لشيفرة Windows NT، هي موضوع معقّد. بشكل عام، تنسب التوافقيّة إلى قدرة نظام تشغيل على تشغيل برامج مكتوبة لأنظمة تشغيل أخرى أو للإصدارات السابقة لنفس النظام. وللنظام Windows NT، يتّخذ مخطّط التوافقيّة عدّة أشكال.

تعريف هذا المخطّط هو مسألة التوافقيّة الثنائيّة مقابل توافقيّة مستوى مصدر البرامج التطبيقيّة. تتحقّق التوافقيّة الثنائيّة عبر تشغيل برنامج تنفيذي بنجاح على نظام تشغيل مختلف. بينما تتطلّب توافقيّة مستوى المصدر إعادة تصريف البرنامج قبل تشغيله على النظام الجديد.

فإذا كان نظام التشغيل الجديد متوافقاً ثنائياً أو متوافق شيفرة المصدر مع نظام موجود، فإنه يعتمد على عدّة أشياء. إن أكثرها أهميّة هو تصميم معالج النظام الجديد. فإذا كان المعالج يستعمل نفس مجموعة التعليمات (وربما مع ملحقات) ونفس حجم عناوين الذاكرة كالمعالج القديم، عندها يمكن تحقيق التوافقيّة الثنائيّة.

إن التوافقيّة الثنائيّة ليست سهلة بين المعالجات ذات التصاميم المختلفة. فكل تصميم لمعالج يحمل معه لغة خاصة بالماكنة. وهذا يعني أنه يمكن تحقيق التوافقيّة الثنائيّة المتقاطعة التصميم فقط إذا زوّد برنامج محاكاة لتحويل مجموعة واحدة من تعليمات الماكنة إلى أخرى. ودون برنامج محاكاة، يجب إعادة تصريف كل البرامج التطبيقيّة التي تنقل من التصميم القديم إلى الجديد وإعادة ربطها (ومن المحتمل إزالة الأخطاء منها).

وعبر إستعمال الأنظمة الفرعيّة المحميّة، يوفّر النظام Windows NT محيط تشغيل للبرامج التطبيقيّة إلى جانب نظام تداخل البرمجة الأولي ــ Winn 32 API. وعند التشغيل على معالج Intel، تزوّد الأنظمة الفرعيّة المحميّة لنظام Windows NT التوافقيّة الثنائيّة مع تطبيقات

Microsoft الموجودة، بما فيها MS-DOS و Windows 16 بت و OS/2 وبرنامج الإدارة LAN. وعلى معالجات MIPS RISC، تتحقّق التوافقيّة الثنائيّة لتطبيقات MS-DOS و Windows 16 بتاً و LAN Managers (باستعمال برنامج محاكاة). كذلك يوفّر النظام Windows NT توافقيّة بمستوى المصدر مع تطبيقات POSIX تتعلّق بأنظمة تداخل نظام التشغيل POSIX المعرّف في مواصفات IEEE رقم 1003.1.

إضافة إلى التوافقيّة مع أنظمة تداخل البرمجة، يدعم النظام Windows NT أنظمة الملفّات الموجودة، بما فيها نظام ملفّات MS-DOS (FAT) ونظام الملفّات المرتفع الأداء OS/2 (HPFS) ونظام ملفّات CD-ROM (CDFS) ونظام الملفّات الجديد Windows NT (NTFS).

5-2-1 الأداء:

لقد كان الهدف التصميمي الأخير لنظام Windows NT تحقيق أداء كبير. تتطلّب التطبيقات الحسابيّة المكثّفة، مثل رزمات الرسوم التخطيطيّة، ورزمات المحاكاة، ورزمات التحليل المالي، معالجة سريعة لتزويد المستعمل بأوقات إستجابة جيّدة. إن الأجهزة السريعة لا تكفي لتحقيق أداء جيّد. لكن، يجب أن يكون نظام التشغيل سريعاً وكافياً أيضاً. لقد كان الأداء الجيّد هدفاً خلال مرحلة تطوير النظام Windows NT. وقد ساعدت العملية التالية تحقيق الهدف:

- صمّم كل مكوّن في نظام Windows NT مع تركيز الإهتمام على الأداء. وقد نفذت إختبارات الأداء على أجزاء النظام المتعلقة به. وتمّ إستمثال إستدعاءات النظام وأخطاء الصفحة وممرّات التنفيذ الحرجة الأخرى بعناية لضمان أكبر سرعة معالجة ممكنة. (راجع الفصل السادس «برنامج إدارة الذاكرة الظاهريّة» والفصل السابع «النواة» للحصول على مزيد من المعلومات).
- يجب على الأنظمة الفرعيّة المحميّة (الملقّمات) التي تنفّذ وظائف نظام التشغيل، الإتصال بتكرار مع بعضها البعض ومع تطبيقات المستضاف. ولضمان عدم قيام هذا الإتصال بتعويق أداء الملقّمات، تمّ شمل آليّة تمرير رسائل عالية السرعة يسمى البرنامج الخدماتي لإستدعاء إجراء محلّي (LPC) كجزء متكامل من نظام التشغيل. (راجع الفصل الرابع «المعالجات والشُعَب» للحصول على مزيد من المعلومات).
- صمّم كل نظام فرعي محميّ يوفّر محيط نظام تشغيل (نظام فرعي محيطي) بعناية لزيادة سرعة ملقّمات النظام الأكثر إستعمالاً إلى الحد الأقصى. (راجع الفصل الخامس «برنامج Windows والأنظمة الفرعيّة المحميّة» للحصول على مزيد من المعلومات).

■ تمّ بناء المكوّنات الحرجة لبراجيّات شبكة النظام Windows NT في قسم نظام التشغيل ذات الأفضليّة لتحقيق أفضل أداء ممكن. ورغم أنها مركّبة بالداخل، يمكن تحميل هذه المكوّنات وإلغاء تحميلها من النظام ديناميكياً. (راجع الفصل التاسع «إنشاء الشبكات» للحصول على مزيد من المعلومات).

3-1 الفريق:

في وقتٍ ما، كان من الممكن لبضعة أشخاص عزل نفسهم والخروج بنظام تشغيل في بضعة أشهر. لكن الأوقات تغيّرت.

إن أنظمة التشغيل الحديثة يجب أن ترضي الأعداد الضخمة من متطلّبات الأجهزة الجديدة، مثل دعم بروتوكولات الشبكة المتعدّدة والمعالجات المتعدّدة وأنظمة الملفّات المتعدّدة والعدد المتزايد من أجهزة الدخل / الخرج. إضافة إلى هذه المتطلّبات الجديدة، يعتبر النظام غير صالح للإستعمال إلا إذا حقّق تعدُّديّة من البراجيّات بما فيها المكتبات ونظام التداخل التخطيطي مع المستعمل والأدوات والتطبيقات ــ دون ذكر المستندات.

إن المجموعة التي صمّمت البرنامج التنفيذي NT وأنظمته الفرعيّة المحميّة الأولى كانت صغيرة ــ حوالي 10 أشخاص في البداية ثم كبرت لتشمل 40 أو 50 شخصاً لاحقاً في المشروع. وسيقدّم هذا الكتاب بعض مصمّمي نظام التشغيل والمستعملين. وهؤلاء الأفراد، رغم قيمتهم الأساسيّة في المشروع، لم يكونوا لينجحوا لولا العديد من الأفراد الآخرين. فالمساهمون في أدوات النظام Windows NT وتطبيقاته ومسيقات الأجهزة وأولئك المسؤولين عن إنفاذ النظام Windows NT ومجموعة من مختبري البراجيات ومديري البرامج وطاقم التسويق والفريق الداعم المؤلّف من مجموعة تزيد عن 200 شخصاً. وفي النهاية، كان إنشاء النظام Windows NT جهداً هائلاً من قِبَل المجموعات المتعدّدة.

4-1 بقيّة الكتاب:

يبدأ الفصل التالي مع نظرة شاملة حول النظام Windows NT والنماذج التي اعتمد عليها وملخّص بمكوّناته. ويشرح كل فصل لاحق مكوّن فردي لنظام التشغيل وخصائصه المهمّة والمزايا البارزة لتصميمه وتفاعلاته مع المكوّنات الأخرى. ويواصل شرح النظام في شكل تصاعدي: فهو يبدأ في الوسط مع المعالجات والكائنات وينتقل بإتجاه الأعلى ليشرح الأنظمة الفرعيّة المحميّة ومحيط API ثم يتعرّج في طريقه إلى الأسفل بإتجاه إدارة الذاكرة والنواة ونظام الدخل / الخرج وإنشاء الشبكات.

2

نظرة شاملة حول النظام

نظام التشغيل هو برنامج حاسوب يوفّر محيطاً تستطيع برامج الحاسوب الأخرى التشغيل عنده، بحيث يتيح لها إستعمال ميزات المعالج وأجهزة الدخل / الخرج مثل الأقراص. إلا أن نظام التشغيل ليس ضرورياً لاستعمال أجهزة الحاسوب. ففي الأيام الأولى لاستعمال الحاسوب، حمّلَ التقنيُّون البرامج في الذاكرة باستعمال أجهزة دخل بدائيّة مثل الأزرار والمفاتيح أو الشريط الورقي. ثم أدخلوا يدوياً عنوان بدء البرنامج ووجّهوا الحاسوب إليه ليبدأ التشغيل. لكنّ مستعملي الحواسيب الآن أصبحوا يستعملون برامج خدماتيّة أكثر تعقيداً.

توفّر أنظمة التشغيل حالياً ملقّمين أساسيّين للمستعملين. الأول، أنه يسهّل إستعمال أجهزة الحاسوب. وهو ينشىء ماكنة «ظاهريّة» تختلف عن الماكنة الفعليّة. وبالطبع، فإن تطوُّر الحواسيب خلال العقدين الأخيرين يعودان في جزء منها إلى نجاح أنظمة التشغيل في حماية المستعملين من مشاكل أجهزة الحاسوب. إضافةً لذلك، لا يحتاج المبرمجون بعد الآن إلى إعادة كتابة برنامج تطبيقي لكل حاسوب يريدون تشغيله عليه.

الثاني، أن نظام التشغيل يشارك موارد الجهاز مع مستعمله. فإحدى أهم الموارد هو المعالج. يقسّم نظام تشغيل متعدّد المهام مثل Windows NT العمل المطلوب تنفيذه خلال المعالجات، بحيث يوفّر لكل ذاكرة معالجة وموارد النظام، وعلى الأقلّ شعبة واحدة من التنفيذ وحدة قابلة للتنفيذ ضمن معالجة. فنظام التشغيل يشغّل شعبة واحدة لفترة قصيرة ثم يحوّل إلى أخرى، حيث يشغل كل شعبة بدورها. وحتى على نظام أحادي المستعمل، فإنّ تعدُّد المهام مساعد للغاية لأنه يمكّن الحاسوب من تنفيذ مهمّتين في نفس الوقت. فمثلاً، يستطيع المستعمل تحرير مستند بينما تتمُّ طباعة مستند آخر في الخلفيّة أو خلال تصريف الحاسوب برنامج كبير. إن كل معالجة تنجز عملها وتبدو للمستعمل وكأن كل البرامج تشتغل في نفس الوقت.

إضافة إلى مشاركة المعالج، يقسّم نظام التشغيل الذاكرة وينظّم الوصول إلى الملفّات والأجهزة. يختلف كل نظام تشغيل في طريقة عرضه الماكنة الظاهريّة إلى المستعملين وفي كيفيّة

تقسيم الموارد عليها. إن الطريقة التي يعتمدها النظام Windows NT لتحقيق ذلك هي موضوع بقيّة هذا الكتاب.

يعالج القسم الأول من هذا الفصل النماذج التي تؤثّر على شكل نظام التشغيل. ويلقي القسم الثاني لمحات على النظام حيث يظهر بنيته الداخليّة. يصف القسم الثالث تصميمين إضافيّين للنظام: التدويل والمناولة الاستثنائية البنيويّة.

1-2 نماذج النظام Windows NT

نظام التشغيل هو برنامج معقّد، عبارة عن طبقات من التفاصيل على تفاصيل. وتنظيم هذه التفاصيل، هذه البتات والباتيات، إلى شكل متماسك هو أحد المهام المهمّة في إنشاء نظام تشغيل جديد. ويحتاج إلى نموذج موحّد لضمان إستيعاب النظام مزاياه المطلوبة دون تعديل أهدافه التصميميّة.

ما هو نموذج نظام تشغيل؟ إن القاموس يعرف الكلمة نموذج على أنها «وصف مؤقّت لنظام أو مبدأ يحتسب كل خصائصه المعروفة». إن نموذج نظام التشغيل هو إطار عمل واسع يوحّد المزايا العديدة والخدمات التي يوفّرها النظام والمهام التي ينفّذها.

لقد استخلص تصميم النظام Windows NT بواسطة دمج عدّة نماذج. فالنظام Windows NT يستعمل نموذجاً مستضافاً / ملقّماً لتوفير محيطات متعدّدة لنظام التشغيل (مبدئياً، Windows و MS-DOS و OS/2 و POSIX) لمستعمليه وهو يستعمل نموذجاً كائنياً لإدارة موارد نظام التشغيل بشكل متماسك وتوفيرها للمستعملين. ويتيح النموذج الثالث، المعالجة المتعدّدة المتناظرة (SMP) للنظام Windows NT تحقيق أداء أقصى من حواسيب متعدّدة المعالجات.

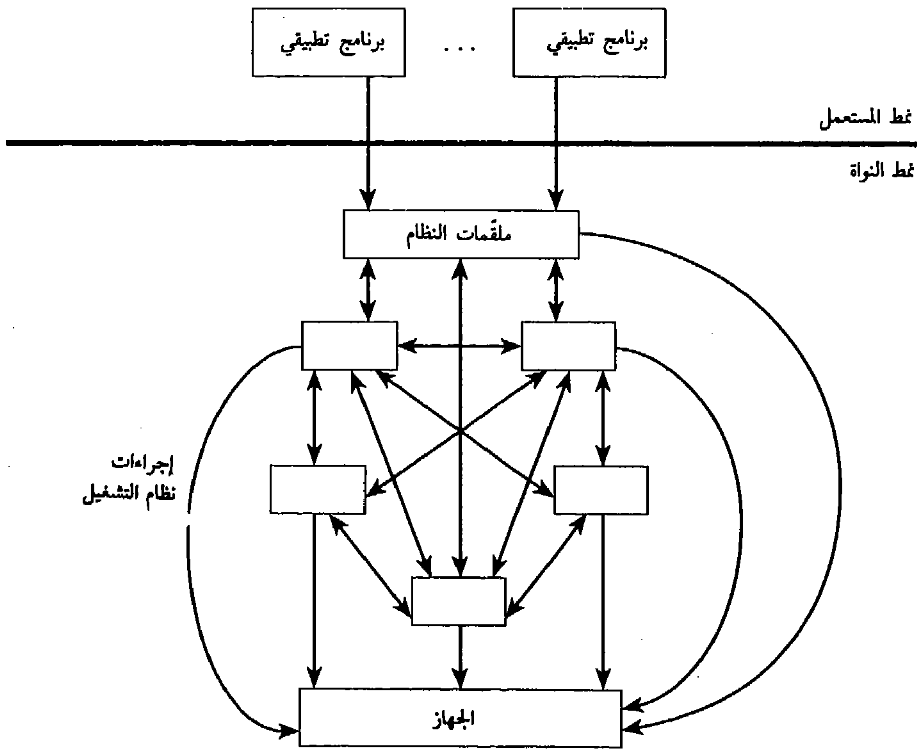
1-1-2 نموذج المستضاف / الملقّم:

يمكن إنشاء شيفرة نظام التشغيل بعدّة طرُق مختلفة. إحدى الطرُق العامّة الإستعمال في أنظمة التشغيل الأصغر مثل MS-DOS، تنظيم نظام التشغيل كمجموعة من الإجراءات، وتتيح لأي إجراء إستدعاء أي إجراء آخر. ولا تجبر هذه البنية الأحاديّة الطبقة إخفاء البيانات في نظام التشغيل، وهي تضمّن الإفتراضات حول كيفيّة ملاءمة النظام سويّة في شيفرة نظام التشغيل. وقد يكون من توسيع مثل هذا النظام عملية صعبة لأن تعديل الأجزاء قد يؤدّي إلى إدخال خلل في الأجزاء غير المتعلّقة بالنظام.

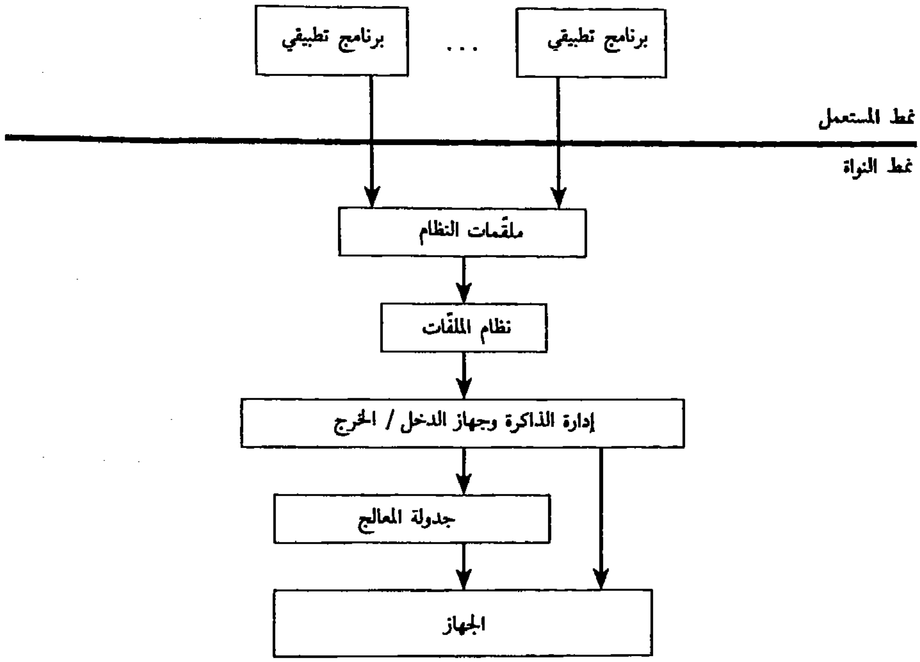
في كل الأنظمة، ما عدا أنظمة التشغيل الأحاديّة الطبقة، تفصل البرامج التطبيقيّة من نظام التشغيل نفسه. وهذا يعني، أن شيفرة نظام التشغيل تشتغل في نمط المعالج بأفضليّة (وهي تسمى نمط النواة في هذا الكتاب). مع الوصول إلى بيانات النظام وإلى الجهاز، والبرامج

التطبيقيّة تشتغل في نمط المعالج بأفضليّة أدنى (تسمّى نمط المستعمل) مع مجموعة محدودة من التداخلات المتوفّرة، ومع وصول محدود إلى بيانات النظام. فعندما يستدعي برنامج في نمط المستعمل ملقّم نظام، يحتجز المعالج الإستدعاء ثم يحوّل شعبة الإستدعاء إلى نمط النواة. وعندما يتمُّ ملقّم النظام، يحوّل نظام التشغيل الشعبة مجدّداً إلى نمط المستعمل ويتيح للمستدعي الإستمرار. تظهر بنية نظام التشغيل الأحادي الطبقة مع نمطي المعالج المستقلّين المستعمل والنواة في الشكل (2-1).

يقسم تصميم بنيوي مختلف نظام التشغيل إلى وحدات ويضعها في طبقات واحدة على الأخرى. توفّر كل طبقة مجموعة من الوظائف التي يمكن للوحدات الأخرى إستدعاءها. وعلى بعض الأنظمة، مثل VAX/VMS أو نظام التشغيل القديم Multics، يقوّم الجهاز بتنظيم الطبقات (باستعمال أنماط المعالج التسلسليّة المتعدّدة). يوضّح الشكل (2-2) على الصفحة التالية إحدى البنيات المرتّبة بطبقات.



الشكل (2-1)
نظام تشغيل أحادي الطبقة



الشكل (2-2)
نظام التشغيل المرتّب بطبقات

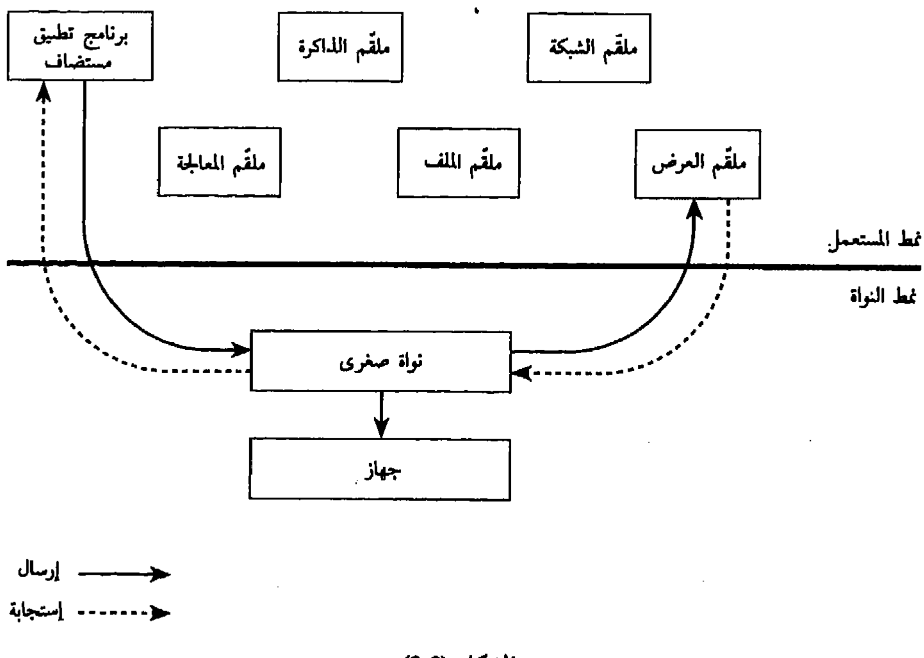
إحدى حسنات بنية نظام تشغيل مرتّب بطبقات هي توفير لكل طبقة شيفرة الوصول إلى تداخلات بمستوى منخفض فقط (وبنـيات البيانات) التي تطلبها، حيث تجد بالتالي من كميّة الشيفرة التي تستخدم طاقة غير محدودة. تتيح هذه البنية أيضاً إزالة العلل من نظام التشغيل بدءاً بالطبقة الأسفل وإضافة طبقة واحدة في كل مرّة إلى أن يعمل كل النظام بشكل صحيح. كذلك، يسهّل إنشاء الطبقات تحسين نظام التشغيل، ويمكن إستبدال طبقة واحدة كاملة دون التأثير على الأجزاء الأخرى من النظام.

الطريقة الثالثة إنشاء بنية نظام تشغيل هي نموذج المستضاف / الملقّم. فالفكرة هي في تقسيم نظام التشغيل إلى عدّة معالجات، وكل منها يستعمل مجموعة واحدة من الملقّمات ــ فمثلاً، ملقّمات الذاكرة وملقّمات إنشاء المعالجة أو ملقّمات جدولة المعالج. يشغّل كل ملقّم في نمط المستعمل حيث ينفذ حلقة تدقّق لجهة طلب مستضاف لإحدى ملقّماته. يطلب المستضاف، الذي يمكن أن يكون مكوّن نظام تشغيل آخر أو برنامج تطبيقي، ملقّماً بواسطة إرسال رسالة إلى الملقّم. يسلّم نظام تشغيل نواة (أو نواة صغري) يشتغل في نمط النواة، الرسالة إلى الملقّم. ينفّذ

الملقّم العمليّة ويرجع نمط النواة النتائج إلى المستضاف في رسالة أخرى، كما هو موضّح في الشكل (2-3).

تؤدّي طريقة المستضاف / الملقّم إلى نظام تشغيل ذات مكوّنات صغيرة وذاتيّ الإحتواء. ولأن كل ملقّم يشتغل في معالجة مستقلّة في نمط المستعمل، فإنه يمكن أن يخفّف ملقّماً واحداً (ويحتمل إعادة بدله) دون أن يؤثّر أو يُشوّه بقيّة نظام التشغيل. إضافة لذلك، يكون تشغيل ملقّمات مختلفة على معالجات مختلفة في حاسوب متعدّد المعالجات أوحتى على حواسيب مختلفة بحيث تجعل نظام التشغيل مناسباً للمحيطات الحاسوبيّة الموزّعة.

إن النموذج النظري المبيّن في الشكل (2-3) هو وصف مثالي لنظام مستضاف / ملقّم حيث تتألّف النواة من برنامج خدماتي يمرّر الرسائل فقط. وفي الواقع، تقع أنظمة المستضاف / الملقّم ضمن طيف، البعض منها يعمل قليلاً جداً في نمط النواة والأخرى تعمل أكثر. فمثلاً، يستخدم نظام التشغيل Mach، مثال معاصر لتصميم المستضاف / الملقّم، نواة دنيا تتألف من جدولة الشعبة وتمرير الرسائل وذاكرة ظاهريّة ومسيقات الأجهزة. وكل شيء آخر، بما فيه تداخلات البرمجة التطبيقيّة (APIs) وأنظمة الملفّات وإنشاء الشبكات، يعمل في نمط المستعمل.



الشكل (2-3)

نظام تشغيل المستضاف / الملقّم

تستعمل بنية النظام Windows NT النموذج المرتّب بطبقات ونموذج المستضاف / الملقّم. يسمى قسم نمط النواة في النظام Windows NT البرنامج التنفيذي NT. وهو يتألف من سلسلة من المكوّنات التي تستخدم إدارة الذاكرة الظاهريّة وإدارة الكائن (المورّد) والدخل / الخرج وأنظمة الملفّات (بما فيها مسيقات الشبكة) والإتصالات داخل المعالجة وأجزاء من نظام الأمان. تتفاعل هذه المكوّنات مع بعضها البعض في طريقة متكاملة بدلاً من ترتيبها بطبقات. ويستدعي كلّ مكوّن المكوّنات الأخرى عبر مجموعة من الروتينات الداخليّة المحدّدة بعناية.

ولكن، يتوفّر نموذج نظام التشغيل المرتّب بطبقات في نظام الدخل / الخرج للبرنامج التنفيذي NT الذي يوصف لاحقاً، وفي الأقسام السفلية من البرنامج التنفيذي NT، نواة NT وطبقة تجريد العتاد (HAL). وترتب المكوّنات الأخرى للبرنامج التنفيذي NT في طبقات على هذين المكوّنين. تنفّذ نواة NT وظائف نظام تشغيل بمستوى منخفض كتلك الموجودة في أنظمة تشغيل المستضاف / الملقّم في النواة الصغرى ـ فمثلاً، جدولة الشعبة والمقاطعة والإرسال الاستثنائي ومزامنة المعالج المتعدّد. وهو يوفّر أيضاً مجموعة من الروتينات والكائنات الأساسيّة التي يستعملها بقيّة البرنامج التنفيذي لتطبيق الإنشاءات بمستوى أعلى. ويوجد تحت النواة مكتبة الربط الدينامي HAL (DLL)، طبقة من الشيفرة التي تحمي النواة وبقيّة البرنامج التنفيذي NT من إختلافات العتاد المحدّد المنصّة. وطبقة HAL تعالج العتاد مباشرة.

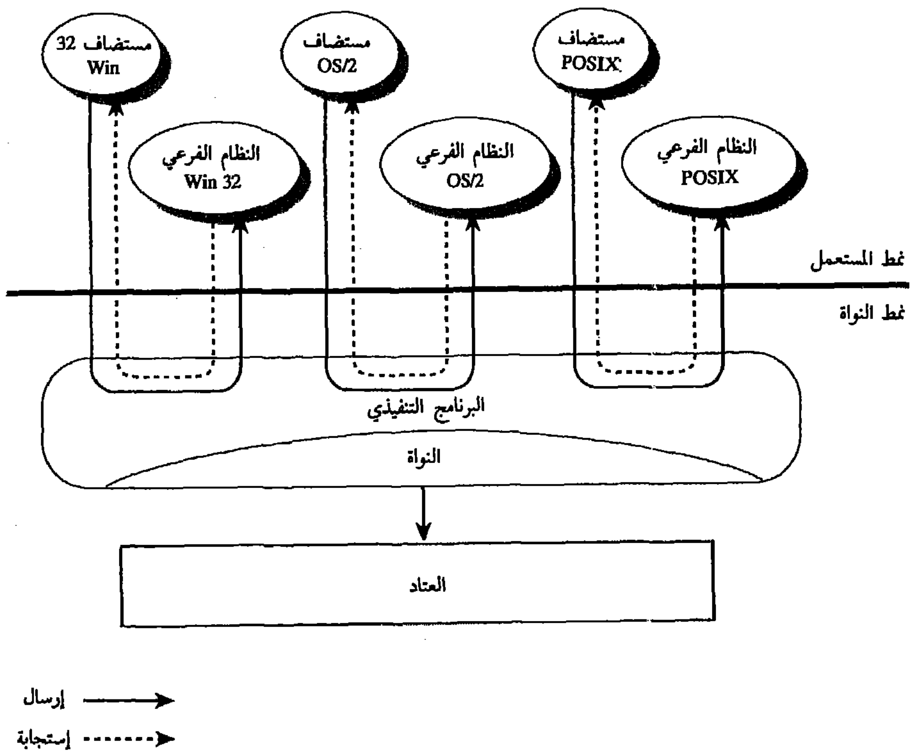
وكما يوضح الشكل (2-4)، يستعمل النظام Windows NT نموذج المستضاف / الملقّم مبدئياً لتوفير APIs والبرامج الخدماتيّة المعتبرة كمحيط نظام تشغيل. ورغم أن النظام الفرعي المحمي Win 32 (الملقّم) يوفّر التداخل مع المستعمل وهو أساسي لتشغيل النظام، فإن الملقّمات الأخرى «تقبس في» البرنامج التنفيذي ويمكن تحميلها على أساس الخلط والمطابقة، مع وجود عدّة ملقّمات أخرى قيد التشغيل. وتتّصل الملقّمات مع معالجات البرنامج التطبيقي عبر برنامج خدماتي لتحرير الرسائل متوفّر في البرنامج التنفيذي NT.

يوفّر إستعمال نموذج المستضاف / الملقّم عدّة فوائد:

- يسهّل النموذج نظام التشغيل الأساسي، البرنامج التنفيذي NT. إن إحدى أهداف النظام Windows NT هو توفير تداخلات API للأنظمة Win 32 و MS-DOS و 16-bit Windows و POSIX و OS/2. ويؤدي نقل كل API إلى ملقّم مستقلّ إلى إزالة التناقضات والإستنساخات من البرنامج التنفيذي، ويتيح إضافة تداخلات API جديدة بسهولة.
- يسهّل النموذج الإعتماديّة. فكلّ ملقّم يشتغل في معالجة مستقلّة، حيث يقسّم إلى ذاكرته وبالتالي تتمّ حمايته من المعالجات الأخرى. إضافة لذلك ولأن الملقّمات تشتغل في نمط

المستعمل، فإنها لا تستطيع الوصول إلى العتاد مباشرة أو تعديل الذاكرة حيث يخزّن البرنامج التنفيذي.

■ يوفّر النموذج نفسه للنموذج الحاسوبي الموزّع. ولأن الحواسيب الموصولة بشبكة تعتمد على نموذج مستضاف / ملقّم وتستعمل الرسائل للإتصال، تستطيع الملقّمات المحليّة إرسال الرسائل بسهولة إلى الماكنات البعيدة نيابة عن تطبيقات المستضاف. ولا تحتاج المستضافات لمعرفة عما إذا كانت بعض الطلبات تُلقَّم محلياً أو عن بُعد.



الشكل (2-4)

بنية المستضاف / الملقّم للنظام Windows NT

2-1-2 نموذج الكائن:

حدّد Bertrand Meyer، في كتابه إنشاء البراجيّات الكائنيّة، خصائص أنظمة التشغيل على أنها برامج «دون سقف» وكما هي الحال مع أنظمة البراجيّات الكبيرة الأخرى، من الصعب

تعريف «برنامج رئيسي» واحد يشغّل نظام تشغيل. لذلك، وعِوَضاً عن محاولة تصميم مثل هذا النظام من الأعلى ونزولاً، تركّز النظريّة الكائنيّة على البيانات التي يجب أن تعالجها البراجيّات لتنفيذ وظيفتها. ولنظام تشغيل، تتّخذ البيانات شكل موارد النظام ـ الملفّات والمعالجات وكتل الذاكرة وما شابه.

إن الهدف الرئيسي لتصميم نظام حول البيانات هو لإنشاء براجيّات سهلة التغيير (ورخيصة). وتصبح أهميّة قدرة التعديل واضحة عند مراجعة الإحصائيّات التي تذكر أن 70 في المئة من كلفة البراجيّات عائدة للصيانة. فصيانة البراجيّات تتضمّن التغييرات مثل إضافة مزايا جديدة وتعديل نسق البيانات وإزالة العِلَل وإستيعاب عتاد جديد.

إحدى الطرق التي تخفض بواسطتها البراجيّات الكائنيّة من التغيير هي عن طريق إخفاء العرض الفعلي للبيانات ضمن الكائنات. والكائن هو بنـية بيانات ذات نسق فعلي مخفيّ خلف تعريف نوع. وهو يتضمّن مجموعة خصائص رسمية (تسمى صفات) ويعالج بواسطة مجموعة من الملقّمات.

ورغم أنه ليس نظام كائني تماماً (مثلما يعرفه Meyer)، يستعمل النظام Windows NT الكائنات لعرض موارد النظام. وتستخدم أي موارد نظام يمكن أن تشارك من قبل أكثر من معالجة واحدة ـ بما فيها الملفّات والذاكرة المشاركة والأجهزة الفعليّة ـ ككائن وتعالج بواسطة ملقّمات الكائن. تخفّف هذه الطريقة من تأثير التغييرات التي ستنفّذ في النظام على مرّ الوقت. وإذا أدّى تغيير العتاد، على سبيل المثال، إلى تغييرات في نظام التشغيل، يجب فقط تغيير الكائن الذي يمثّل مورد العتاد والملقّمات التي تعمل على الكائن. وتبقى الشيفرة التي تستعمل الكائن بالكاد كما هي. وبشكل مشابه، عندما يحتاج النظام لدعم موارد جديدة، يتمّ إنشاء كائناً جديداً ويُضاف إلى النظام دون تغيير الشيفرة الموجودة.

إضافة لتحديد تأثيرات التغيير، فإن إنشاء نظام تشغيل كائني يتّصف بميزات مميّزة:

- يستطيع نظام التشغيل الوصول إلى موارده ومعالجتها بشكل متناسق. وهو ينشىء كائن حدث ويحذفه ويعود إليه بنفس الطريقة التي يعتمدها مع كائن معالجة: بإستعمال مقابض الكائن. ولأن كل مورد هو كائن، يتمّ تعقّب إستعمال الموارد بواسطة مراقبة إنشاء الكائنات وإستعمالها.
- يبسّط الأمان لأن كل الكائنات محميّة بنفس الطريقة. وعندما سيحاول شخص ما الوصول إلى كائن، يتدخّل نظام الأمان ويتحقّق من صلاحيّة التشغيل دون إعتبار لجهة كون الكائن معالجة أو مقطع من ذاكرة مشاركة أو منفّذ إتصال.
- توفّر الكائنات نموذجاً مناسباً ومتناسقاً لمشاركة الموارد بين معالجتين أو أكثر. تستعمل مقابض

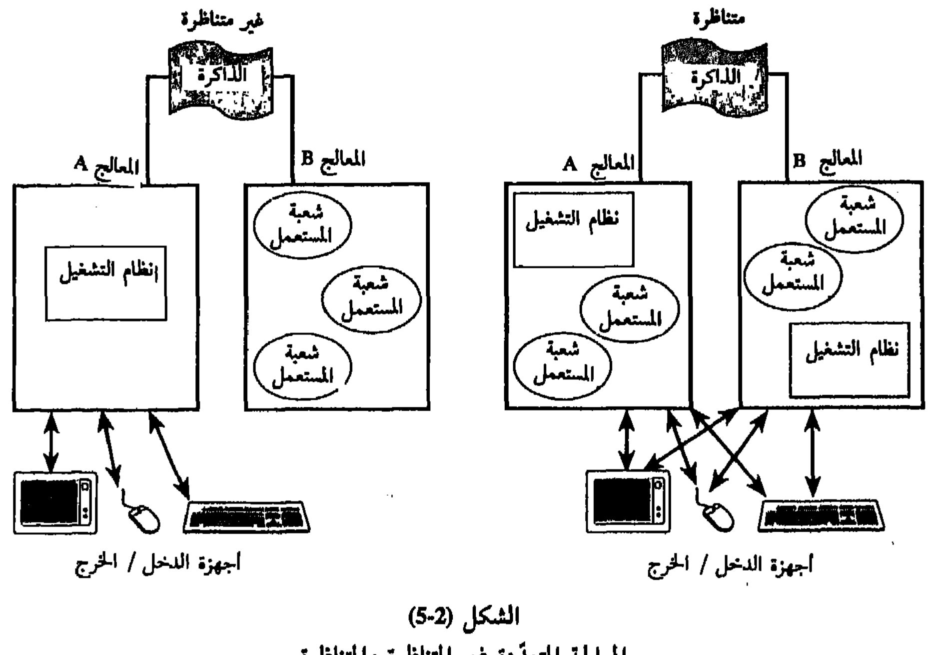
الكائن لمناولة كل أنواع الكائنات. وتشارك معالجتا كائن عندما تفتحان مقبضاً إليه. ويستطيع نظام التشغيل تعقُّب عدد المقابض المفتوحة لكائن تحديد إذا ما زال الكائن قيد الإستعمال. يمكن أن يقوم نظام التشغيل بعد ذلك بحذف الكائن عند عدم إستعماله.

يصف الفصل الثالث، «برنامج إدارة الكائن وأمان الكائن»، برنامج إدارة الكائن ومكوّن البرنامج التنفيذي NT الذي يستخدم كائنات النظام Windows NT ويديرها.

2-1-3 المعالجة المتعدّدة المتناظرة:

المهام المتعدّدة هي طريقة نظام التشغيل في مشاركة معالج واحد ضمن شِعَب متعدّدة من التنفيذ. ولكن، عندما يحتوي الحاسوب على أكثر من معالج واحد، يجب تحسين نموذج المهام المتعدّدة إلى نموذج المعالجة المتعدّدة. يستطيع الحاسوب الذي يحتوي على معالجين تنفيذ شعبتين في نفس الوقت. وبالتالي وحيث تنفذ نظام تشغيل متعدّد المهام شِعَب متعدّدة في نفس الوقت. فإن نظام تشغيل متعدّد المعالجة يقوم بذلك أيضاً حيث ينفذ شعبة واحدة على كل من معالجيه.

تصنّف أنظمة التشغيل المتعدّدة المعالجة ضمن فئة من فئتين، حيث تدعم المعالجة غير المتناظرة أو المتناظرة. كما يوضّح في الشكل (2-5).



الشكل (2-5)
المعالجة المتعدّدة غير المتناظرة والمتناظرة

تنتمي أنظمة التشغيل بمعالجة متعدّدة غير متناظرة (ASMP) نموذجياً نفس المعالج (A، على سبيل المثال) لتنفيذ شيفرة نظام التشغيل بينما تنفّذ المعالجات الأخرى وظائف المستعمل فقط. ولأن شيفرة نظام التشغيل تعمل على معالج واحد، فإن أنظمة التشغيل ASMP سهلة الإنشاء نسبياً عن طريق تمديد أنظمة التشغيل الأحاديّة المعالج الموجودة. وتتناسب أنظمة التشغيل ASMP للتشغيل على عتاد غير متناظر، مثل معالج مع معالج فرعي مثبّت أومعالجَين لا يتشاركان كل الذاكرة المتوفّرة. لكن يصعب جعل أنظمة التشغيل ASMP نقّالة. فالعتاد من البائعين المختلفين (وحتى الإصدارات المختلفة للعتاد من نفس البائع) يختلف في نوعه ودرجة اللاتناظر. فيجب إما حصول البائعين على عتاد يناسب أنظمة تشغيل معيّنة أو يجب إعادة كتابة نظام التشغيل لكلّ منصّة عتاد.

تتيح أنظمة التشغيل بمعالجة متعدّدة متناظرة (SMP)، بما فيها Windows NT، تشغيل نظام التشغيل على أي معالج حرّ أوعلى كل المعالجات في نفس الوقت، ومشاركة الذاكرة فيما بينها. وتستخدم هذه الطريقة طاقة المعالجات المتعدّدة لأن نظام التشغيل نفسه يستطيع إستعمال نسبة كبيرة من وقت المعالجة في الحاسوب، وفقاً للبرامج التطبيقيّة التي تشغلها. إن تشغيل نظام التشغيل على معالج واحد فقط يضيف المعالج المستعمل ويترك الأخرى دون إستعمال ويخفّف من إخراج النظام. وكلما إزداد عدد المعالجات على النظام تصبح نشاطات نظام التشغيل محشورة. وإضافة لموازنة حمل النظام، تخفّض أنظمة SMP وقت التعطّل لأن شيفرة نظام التشغيل تستطيع التنفيذ على معالجات أخرى إذا تعطّل معالج واحد. وأخيراً ولأن العتاد المتناظر مستخدم بشكل مشابه من بائع إلى آخر، من الممكن إنشاء نظام تشغيل SMP نقّال.

وبعكس أنظمة ASMP، تُصمَّم أنظمة SMP وتُكتب عادة من الأسفل وصعوداً لأنه يجب أن تلازم خطوط توجيه الشيفرة من قرب لضمان التشغيل الصحيح. أما محتويات الموارد ومواضيع الأداء الأخرى فهي أكثر تعقيداً في أنظمة المعالجة المتعدّدة مما هي في أنظمة التشغيل العادية ويجب أن تعتبر في مرحلة تصميم النظام.

يشمل النظام Windows NT عدّة مزايا خاصة بنجاحه كنظام تشغيل متعدّد المعالجة:

- القدرة على تشغيل شيفرة نظام تشغيل على أي معالج متوفر وعلى المعالجات المتعدّدة في نفس الوقت. وباستثناء مكوّنها النواة، والتي تتناول جدولة الشعَب والمقاطعات، يمكن أن يستولي على كل شيفرات نظام التشغيل (جعلها تتخلّى عن معالج واحد) عند ضرورة معالجة شعبة ذات أولويّة أعلى.

- شعَب تنفيذ متعدّدة ضمن معالجة واحدة. تتيح الشعَب لمعالجة واحدة تنفيذ أجزاء مختلفة من برنامجها على عدّة معالجات في نفس الوقت.
- معالجات الملقّم التي تستعمل شعَباً متعدّدة لمعالجة الطلبات من أكثر من مستضاف واحد في نفس الوقت.
- آلية مناسبة لمشاركة الكائنات بين المعالجات وقدرات الإتصال داخل المعالجة المرنة، بما فيها الذاكرة المتشاركة والبرنامج الخدماتي المستمثل لتمرير الرسائل.

يتمُّ شرح المعالجات والشعَب في الفصل الرابع، «المعالجات والشعَب» وتشرح ملقّمات Windows NT في الفصل الخامس «Windows والأنظمة الفرعيّة المحميّة».

2-2 بنية النظام Windows NT

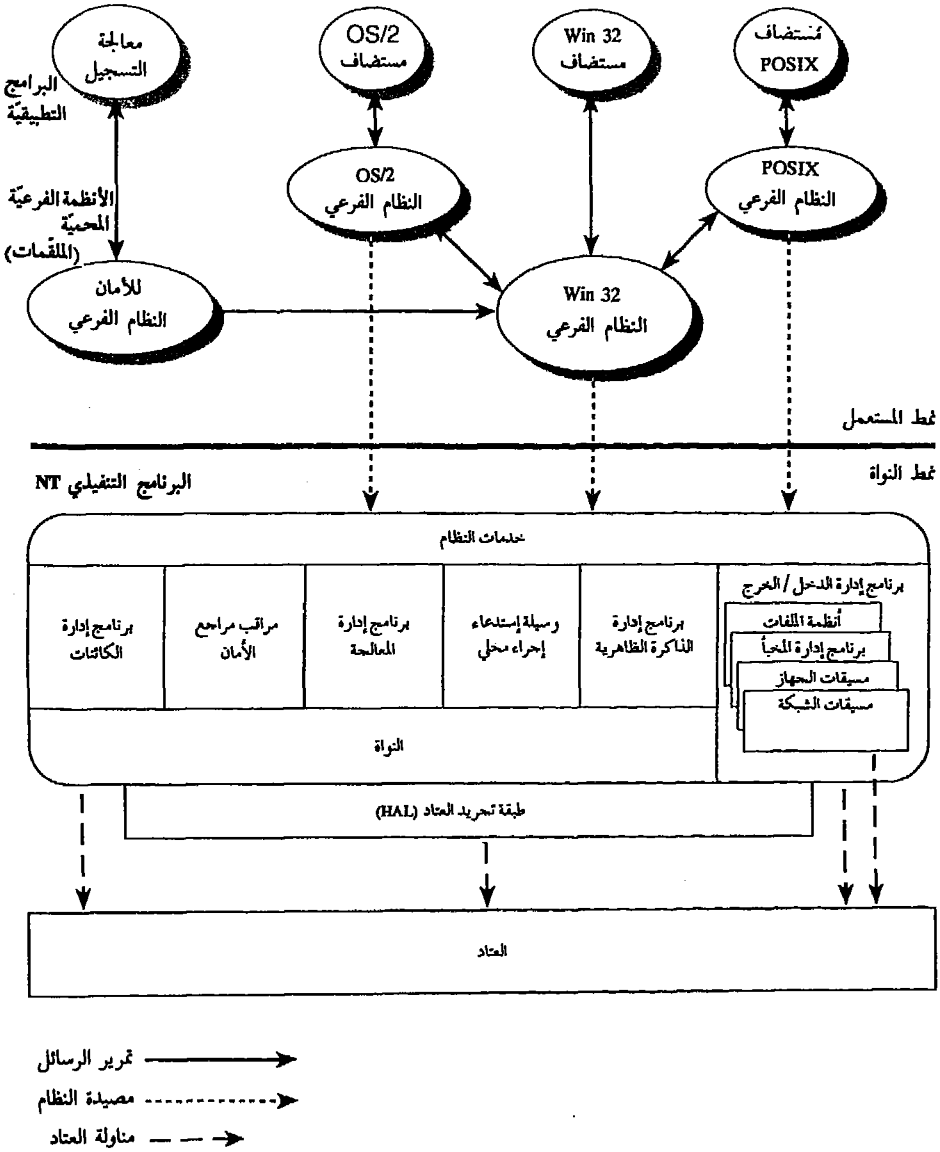
يمكن تقسيم بنية النظام Windows NT إلى قسمين: قسم نمط المستعمل من النظام (الأنظمة الفرعيّة المحميّة للنظام Windows NT) وقسم نمط النواة (البرنامج التنفيذي NT). يبيّن رسم توضيحي مفصّل لنظام Windows NT في الشكل (2-6) على الصفحة التالية.

تسمّى ملقّمات النظام Windows NT الأنظمة الفرعيّة المحميّة لأن كل منها يستقرُّ في معالجة مستقلّة ذات ذاكرة محميّة من المعالجات الأخرى بواسطة نظام الذاكرة الظاهريّة للبرنامج التنفيذي NT. ولأن الأنظمة الفرعيّة لا تتشارك والذاكرة تلقائياً، فإنها تتصل بواسطة تمرير الرسائل. تمثّل الخطوط الثخينة في الشكل (2-6) المسارات التي تتّخذها الرسائل بين المستضافات والملقّمات أو بين ملقّمين. تمرّر كل الرسائل عبر البرنامج التنفيذي، لكن للتسهيل، لا تبيّن هذه المسارات في الشكل.

وكما ذُكر سابقاً، فإن البرنامج التنفيذي NT هو محرّك نظام تشغيل قادر على دعم أي عدد من معالجات الملقّم. فالملقّمات تقدّم مستعملي البرنامج التنفيذي NT وتداخلات البرمجة وتوفّر محيطات التنفيذ لأنواع التطبيقات المختلفة. إن القسمين التاليين يشرحان بتفصيل بُنية النظام Windows NT.

1-2-2 الأنظمة الفرعيّة المحميّة:

كما يشير التعبير «ملقّم»، يوفّر كل نظام فرعي محمي تداخل API تستطيع البرامج إستدعاءه. وعندما يستدعي برنامج تطبيقي روتين API، ترسل رسالة إلى الملقّم الذي يطبّق روتين API عبر وسيلة إستدعاء إجراء محلي (LPC) للبرنامج التنفيذي NT، وهي آلية مستمثلة لتمرير الرسائل المحليّة. يستجيب الملقّم عن طريق إرسال رسالة إلى المستدعي.



الشكل (2-6)

مخطّط مراحل النظام Windows NT.

يحتوي النظام Windows NT على نوعين من الأنظمة الفرعيّة المحميّة: الأنظمة الفرعيّة المحيطيّة، والأنظمة الفرعيّة المتكاملة. فالنظام الفرعي المحيطي هو ملقّم نمط المستعمل الذي يوفّر روتين API محدّد إلى نظام تشغيل. وعندما يستدعي برنامجي تطبيقي روتين API، يسلّم

طلب الاستدعاء عبر الوسيلة LPC إلى النظام الفرعي المحيطي. ينفّذ النظام الفرعي المحيطي روتين API ويرجع النتيجة إلى معالجة البرنامج التطبيقي عند طريق إرسال إستدعاء LPC آخر.

إن أهم نظام فرعي محيطي للنظام Windows NT هو النظام الفرعي WIN 32 الذي يوفّر روتين API لنظام 32-bit Windows من Microsoft إلى البرامج التطبيقيّة. إضافة لذلك، يوفّر النظام الفرعي المحيطي Win 32 نظام التداخل التخطيطي مع المستعمل للنظام Windows NT ويتحكّم بكل دخل المستعمل وخرج البرنامج التطبيقي. كذلك يزوّد النظام Windows NT النظام الفرعي المحيطي POSIX والنظام الفرعي المحيطي OS/2 والنظام الفرعي المحيطي 16-bit Windows والنظام الفرعي المحيطي MS-DOS. (لا يظهر هذان الأخيران في الشكل (2-6)). وهذه الأنظمة الفرعيّة توفّر روتينات API لكنّها تستعمل النظام الفرعي Win 32 لإستلام دخل المستعمل ولعرض الخرج.

أما الأنظمة الفرعيّة المحميّة المتبقّية ــ الأنظمة الفرعيّة المتكاملة ــ فهي ملقّمات تنفّذ الوظائف المهمّة لنظام التشغيل. ولقد ظهرت العديد من الأنظمة الفرعيّة المتكاملة واختفت خلال تطوير النظام Windows NT، لكن بقي نظام فرعي متكامل واحد: النظام الفرعي للأمان. يشتغل النظام الفرعي للأمان في نمط المستعمل ويسجّل سياسات الأمان قيد التأثير على الحاسوب المحلّي. فمثلاً، فإنه يتعقّب حسابات المستعمل ذات الأفضليّة الخاصة وموارد النظام المدقّقة للوصول والحاجة لإصدار إنذارات تدقيق أو رسائل تدقيق. إضافة لذلك، يحافظ النظام الفرعي للأمان على قاعدة بيانات تتعلّق بحسابات المستعملين بما في ذلك إسم الحساب وكلمات السرّ والمجموعة التي ينتمي إليها المستعمل لأغراض الأمن وأي تفضيلات خاصة يمتلكها المستعمل. وهو يقبل أيضاً معلومات التسجيل ويحفز توثيق التسجيل.

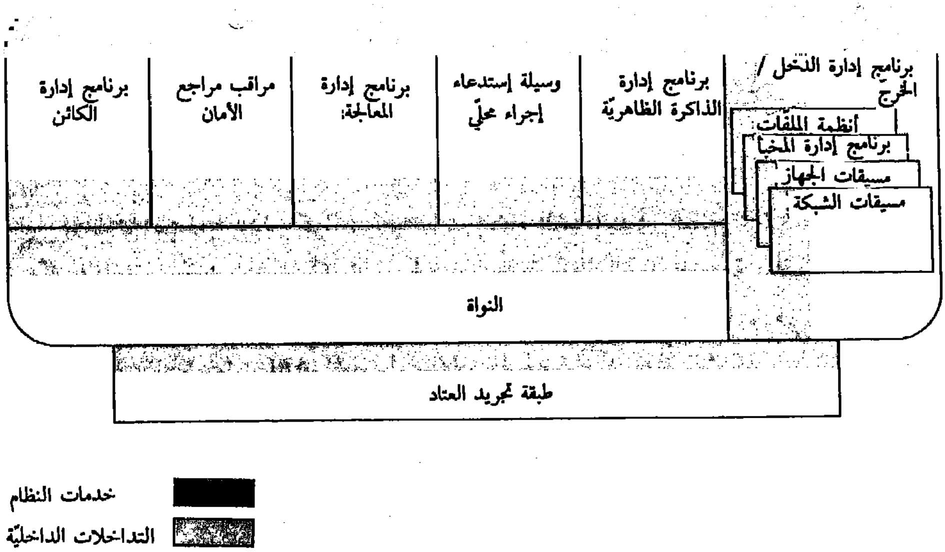
كذلك تستخدم عدّة مكوّنات لبرامجيّات إنشاء الشبكات في النظام Windows NT كأنظمة فرعيّة متكاملة، حيث يشرح مكوّنان منها هنا: خدمة محطة العمل وخدمة الملقّم. كل من هذه الخدمات، كما تسمى الأنظمة الفرعيّة لإنشاء الشبكات في غالب الأحيان، هي معالجة في نمط المستعمل تستخدم روتين API للوصول وإعادة الموجّه الجديد وملقّم شبكة برنامج إدارة LAN على التوالي. بأن الموجّه الجديد هو مكوّن الشبكة المسؤول عن إرسال (التوجيه بوجهة جديدة) طلبات الدخل / الخرج عبر شبكة عندما لا يكون الملفّ أو الجهاز الواجب الوصول إليه محلّي. يستقرُّ الملقّم على الماكنة البعيدة ويستلم مثل هذه الطلبات البعيدة. ويستعمل كلّاً من الموجّه الجديد لبرنامج إدارة LAN وملقّم برنامج إدارة LAN كمسيقات نظام الملفّات ــ أي، كجزء من نظام الدخل / الخرج NT الذي يوصَف لاحقاً.

2-2-2 البرنامج التنفيذي:

البرنامج التنفيذي NT هو قسم نمط النواة للنظام Windows NT، وباستثناء التداخل مع المستعمل، فهو نظام تشغيل كامل بنفسه. يتألف البرنامج التنفيذي من سلسلة من المكوّنات، كل منها يستعمل مجموعتين من الوظائف: خدمات النظام التي تستطيع الأنظمة الفرعيّة المحيطة والمكوّنات الأخرى للبرنامج التنفيذي إستدعاءها، والروتينات الداخليّة، المتوفّرة فقط للمكوّنات الموجودة ضمن البرنامج التنفيذي. توضح التداخلات في الشكل (2-7).

رغم أن البرنامج التنفيذي يوفّر خدمات نظام مشابهة لخدمات API، إلا أنه يختلف مبدئياً عن الأنظمة الفرعيّة المحيطيّة. فهو لا يشتغل بشكل متواصل في معالجة خاصة به، لكنه يشتغل في سياق معالجة موجودة عن طريق الاستيلاء على شعبة تنفيذ عند حصول أحداث مهمّة في النظام. فمثلاً، عندما تستدعي شعبة خدمة نظام ويتمّ إحتجازها من قِبَل المعالج أو عندما يقاطع جهاز خارجي المعالج، تستولي نواة NT على التحكّم بالشعبة التي كانت تشتغل. وتستدعي النواة شيفرة النظام المناسبة لمناولة الحدث وتنفيذه ثم إرجاع الحكم إلى الشيفرة التي كانت تنفّذ قبل المقاطعة.

تحافظ مكوّنات البرنامج التنفيذي على إستقلاليتها عن بعضها البعض، حيث تنشىء كل منها بنيات بيانات النظام التي تحتاجها وتتناولها. ولأن التداخلات بين المكوّنات محكومة بعناية،



الشكل (2-7)
تداخلات النظام

يمكن إزالة مكوّن بالكامل من نظام التشغيل وإستبداله بآخر يعمل بشكل مختلف. وطالما أن الإصدار الجديد يستعمل كل خدمات النظام والتداخلات الداخليّة بشكل صحيح، يشتغل نظام التشغيل كما في السابق. إن المحافظة على نظام التشغيل هي مهمّة سهلة أيضاً لأن مكوّنات البرنامج التنفيذي NT تتفاعل بطرُق متوقّعة.

يوجد أدناه مسؤوليات مكوّنات البرنامج التنفيذي:

- **برنامج إدارة الكائن**: ينشىء كائنات البرنامج التنفيذي NT ويديرها ويحذفها ويجرّد أنواع البيانات المستعملة لعرض موارد نظام التشغيل.
- **مراقب مراجع الأمان**: يحفز سياسات الأمان على الحاسوب المحلي. وهو يحمي موارد نظام التشغيل وينفّذ عمليّات حماية وتدقيق الكائن في وقت التشغيل.
- **برنامج إدارة المعالجة**: ينشىء المعالجات والشعَب وينهيها. وهو يوقف تنفيذ الشعَب ويعاوِد تشغيلها ويخزّن المعلومات المتعلّقة بشعَب ومعالجات NT ويستردّها.
- **وسيلة إستدعاء إجراء محلي (LPC)**: يمرّر الرسائل بين معالجة مستضاف ومعالجة ملقّم على نفس الحاسوب. إن LPC هو إصدار مَرِن مستمثل لاستدعاء إجراء بعيد (RPC) وهو وسيلة للإتصالات قياسي ـ صناعي لمعالجات المستضاف والملقّم عبر شبكة. (راجع الفصل التاسع، «إنشاء الشبكات»، للحصول على مزيد من المعلومات).
- **برنامج إدارة الذاكرة الظاهريّة (VM)**: يستعمل الذاكرة الظاهريّة، مخطّط إدارة ذاكرة يوفّر فسحة عنوان كبيرة خاصة لكل معالجة ويحمي كل فسحة عنوان معالجة من المعالجات الأخرى. وعندما يكون إستعمال الذاكرة كثيفاً جداً، ينقل برنامج إدارة VM محتويات الذاكرة المنتقاة إلى قرص ويُعيد تحميل المحتويات عند إستعمالها مجدّداً وهو ما يُعرَف بتعيين الصفحات.
- **النواة**: تستجيب للمقاطعات والاستثناءات وتُجَدوِل الشعَب للتنفيذ وتزامن نشاطات المعالجات المتعدّدة وتزوّد مجموعة من الكائنات الفرديّة والتداخلات التي تستعملها بقيّة البرنامج التنفيذي NT لتطبيق الكائنات ذات المستوى الأعلى.
- **نظام الدخل / الخرج**: يتألّف من مجموعة مكوّنات مسؤولة عن معالجة الدخل من تسليم الخرج إلى مجموعة أجهزة متنوّعة. يشمل نظام الدخل / الخرج المكوّنات الفرعيّة التالية:
  - **برنامج إدارة الدخل / الخرج**: يستخدم البرامج الخدماتيّة للدخل / الخرج المستقلّة عن الجهاز ويحقّق نموذجاً لدخل / خرج البرنامج التنفيذي NT.

- ☐ **أنظمة الملفّات:** مسيقات NT التي تقبل طلبات الدخل / الخرج الملفاتية وتترجمها إلى طلبات دخل / خرج خاصة بجهاز معيّن.
- ☐ **الموجّه الجديد للشبكة وملقّم الشبكة:** مسيقات نظام الملفّات التي ترسل طلبات دخل / خرج بعيدة إلى ماكنة على الشبكة وتستلم مثل هذه الطلبات على التوالي.
- ☐ **مسيقات جهاز البرنامج التنفيذي NT:** مسيقات بمستوى منخفض تتناول العتاد مباشرة لكتابة الخرج أو لإستلام الدخل من جهاز فعلي أو شبكة.
- ☐ **برنامج إدارة المخبأ:** يحسّن أداء دخل / خرج الملفّ عن طريق تخزين معلومات القرص المقروء مؤخراً في ذاكرة النظام. يستعمل برنامج إدارة المخبأ البرنامج الخدماتي لتصميم صفحات برنامج إدارة VM ليكتب تلقائياً التعديلات إلى القرص في الخلفيّة.

- ■ **طبقة تجريد العتاد (HAL):** يضع طبقة من الشيفرة بين البرنامج التنفيذي NT ومنصّة العتاد حيث يشتغل النظام Windows NT. وهي تخفي تفاصيل العتاد مثل تداخلات الدخل / الخرج ووحدات التحكّم بالمقاطعة وآليّات الإتصال للمعالج المتعدّد. وعِوَضاً عن الوصول مباشرة إلى العتاد، تحافظ مكوّنات البرنامج التنفيذي NT على تنقليّة قصوى عن طريق إستدعاء روتينات HAL عندما تحتاج لمعلومات تتعلّق بالمنصّة.

إن النظام Windows NT هو نظام تشغيل نقّال، مصمّم ليحدّ من كميّة الشيفرة التي تعتمد على تصميم عتاد معيّن. لكن يحتاج لبعض الشيفرات الخاصة بالمعالج (مثلاً Intel 486 أو MIPS R4000) وهي تقع في الطبقات السفلى من نواة NT مع وجود أقسام صغيرة في برنامج إدارة VM. تخفي هذه المكوّنات وخاصة نواة NT إختلافات المعالج عن بقيّة نظام التشغيل.

تقع شيفرة المنصّة، أي الشيفرة التي تعتمد على تطبيق مصنّع خاص للحاسوب MIPS R4000، مثلاً ــ في طبقة HAL وهي تتوفّر من قِبَل مصنّعي الحاسوب الإفرادي. تحتوي مسيقات الجهاز شيفرة خاصة بالجهاز لكنها تتجنّب شيفرات المعالج وشيفرات المنصّة عن طريق إستدعاء روتينات النواة NT وروتينات HAL.

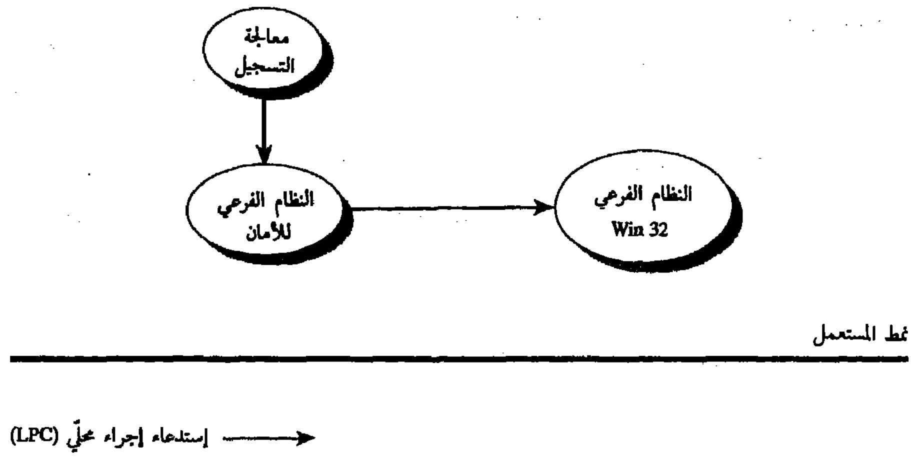
3-2-2 جولة موجزة:

باستثناءات بسيطة، لا يبدو النظام Windows NT وكأنه نظام تشغيلي جديد من وجهة نظر المستعمل. فهو يبدو كالنظام Windows ويشغل برامج Windows. لكن تحت التداخلات مع المستعمل، فإن الإختلاف واضح. توفّر الأقسام التالية جولة سريعة عن كيفيّة تلائم الأجزاء

المختلفة للنظام Windows NT، بدءاً من التداخل مع المستعمل نزولاً إلى البرنامج التنفيذي NT.

1-3-2-2 دورة التسجيل:

إن النظام Windows NT هو نظام تشغيل آمن يتطلّب من كل مستعمل إنشاء حساب والتسجيل في ذلك الحساب قبل تمكينه من الوصول إلى النظام. يتعلّق بكل حساب مستعمل إستمثال أمان، وهو مجموعة من معلومات الأمان المخزّنة في قاعدة بيانات النظام. تستخدم الأنظمة الفرعيّة للأمان هذه المعلومات للتحقّق من المستعمل الذي يطلب إستعمال النظام. تميّز مكوّنات النظام المشمولة في عمليّة التسجيل في الشكل (2-8).



الشكل (2-2)
التسجيل

تنتظر معالجة نظام أمان، تسمى معالجة التسجيل، دخل المستعمل. يمكن تنشيط عدّة معالجات تسجيل، وكل واحدة تراقب فئة منفصلة من أجهزة التسجيل. مثلاً: توليفة لوحة المفاتيح / الماوس أو توصيل شبكة. تكشف شعبة في المعالجة محاولة المستعمل الوصول إلى النظام وتحث المستعمل لإدخال إسم الحساب وكلمة سرّ.

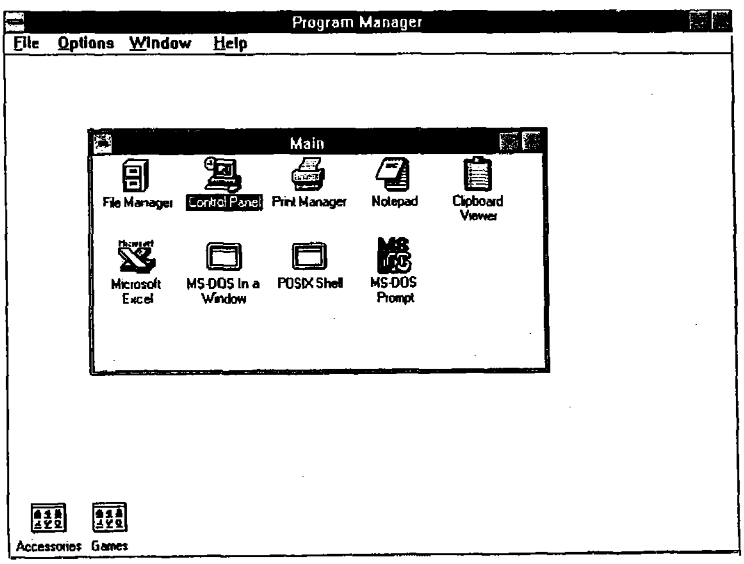
من هناك، تمرّر معالجة التسجيل معلومات المستعمل إلى النظام الفرعي للأمان الذي يدقّق بالمعلومات مقابل قاعدة بيانات الأمان. فإذا كان التسجيل صحيحاً، ينشىء النظام الفرعي كائناً يعرّف بشكل فريد هذا المستعمل في كل العمليّات اللاحقة. ويكون الكائن،

الذي يسمى صفة الوصول، مفتاح الأمان في النظام Windows NT : فهو يحدّد موارد النظام التي تستطيع شعَب المستعمل الوصول إليها.

بعد التأكّد من المستعمل، ينشىء النظام الفرعي للأمان معالجة، تثبت صفة وصول المستعمل إليها ثم يمرّر المعالجة إلى النظام الفرعي Win 32 الذي يشغّل برنامج إدارة البرامج Win 32 Program Manager في فسحة عنوان المعالجة وبذلك يكون المستعمل قد أتمّ دورة تسجيل. يدعم النظام Windows NT التسجيلات المحليّة والبعيدة ويحتمل أن تحتوي ماكنة ملقّم تشغّل النظام Windows NT على عدّة دورات تسجيل نشطة في نفس الوقت.

وحالما يسجّل مستعمل متفاعل داخلياً بنجاح على النظام Windows NT، يتحكّم النظام الفرعي Win 32 بالشاشة. وفي إصداره الأول، يشبه النظام Windows NT النظام Windows 3.1 ويتوافق معه كما يظهر في الشكل (2-9).

وبواسطة النظام Windows NT، يستطيع المستعمل تشغيل برامج Win 32 وبرامج 16-bit Windows وكذلك برامج MS-DOS و OS/2 و POSIX.



الشكل (2-9)

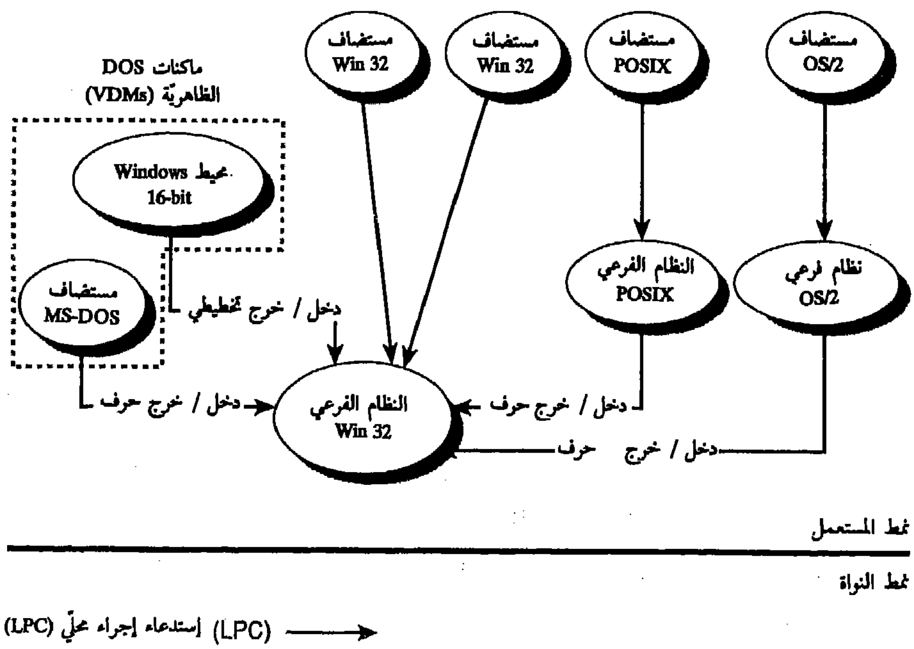
التداخل مع المستعمل في النظام Windows NT

2-3-2-2 الأنظمة الفرعيّة للمحيط:

يوفّر النظام الفرعي للمحيط Win 32 التداخل مع المستعمل في النظام Windows NT وإضافة إلى التحكّم بالعرض، فإنه يتحكّم بلوحة المفاتيح أيضاً والماوس وأجهزة الدخل الأخرى المثبّتة بالماكنة. وإضافة لذلك، فهو ملقّم للبرامج التطبيقيّة Win 32 ويستخدم روتينات Win 32 API.

ليست كل التطبيقات هي تطبيقات Win 32، ولا يتحكّم النظام الفرعي Win 32 بتنفيذ التطبيقات غير العائدة إلى Win 32. فعندما يشغل المستعمل برنامجاً تطبيقياً لا يتعرّف إليه النظام الفرعي Win 32، يحدّد النظام الفرعي نوع البرنامج التطبيقي ثم إما يستدعي نظام فرعي آخر لتشغيل البرنامج التطبيقي أو يستدعي الشيفرة لتحفيز محيط MS-DOS لكي يشغّل البرنامج التطبيقي (راجع الشكل (2-10)).

يزوّد كلاً من الأنظمة الفرعيّة للمحيط روتينات API تستعملها تطبيقات المستضاف العائدة لها. فمثلاً، يزوّد النظام الفرعي Win 32 روتينات 32-bit Windows API ويزوّد النظام الفرعي OS/2 روتينات OS/2 API. ولا يمكن خلط التطبيقات ومحافظة روتينات API من الأنظمة



الشكل (2-10)
الأنظمة الفرعيّة للمحيط وتطبيقات المستضاف

الفرعيّة المختلفة لأن كل نظام فرعي للمحيط يعمل بطريقة مختلفة. ولا يترجم مقبض ملف تم إنشاؤه بواسطة النظام الفرعي Win 32 إلى النظام الفرعي POSIX على سبيل المثال. إضافة لذلك، لن تعمل هذه التطبيقات الهجينة على أنظمة التشغيل MS-DOS / Windows أو POSIX أو OS/2.

تزوّد محاكاة MS-DOS و 16-bit Windows من قِبَل نظام فرعي لمحيط يسمى ماكنة DOS الظاهريّة (VDM) التي تزوّد محيط كامل الماكنة MS-DOS. تشغّل تطبيقات MS-DOS و 16-bit Windows ضمن سياق معالجات VDM التي تختلف عن الأنظمة الفرعيّة للمحيط الأخرى لجهة إمكانيّة تشغيل عدّة معالجات VDM في نفس الوقت. (راجع الفصل الخامس، «Windows والأنظمة الفرعيّة المحميّة» للحصول على مزيد من المعلومات).

ولأن النظام الفرعي Win 32 يتناول كل خرج الشاشة، يجب على الأنظمة الفرعيّة للمحيط الأخرى توجيه خرج تطبيقاتها إلى النظام الفرعي Win 32 لعرضها. وتترجم VDM التي تشغّل تطبيقات 16-bit Windows إستدعاءات خرج التطبيقات إلى إستدعاءات Win 32 وترسلها في رسالة إلى النظام الفرعي Win 32 لعرضها. وكذلك توجّه الأنظمة الفرعيّة OS/2 و POSIX وأيّ VDMs تشغّل تطبيقات MS-DOS خرج نمط أحرف التطبيقات إلى النظام Win 32 الذي يعرض الخرج من نوافذ نمط الأحرف، التي تسمى كونسولات.

يستطيع نظام فرعي لمحيط دعم عدّة تطبيقات مستضاف. ويتابع كل نظام فرعي تعقُّب مستضافاته ويحافظ على أية معلومات شموليّة تتشاركها كل تطبيقات المستضاف. ورغم إمكانيّة إشتغال عدّة أنظمة فرعيّة وماكنات VDM، فإن Win 32 هو النظام الفرعي للمحيط الوحيد الذي يجعل نفسه مرئياً. وبالنسبة للمستعمل، يظهر وكأن النظام Windows يشغّل كل التطبيقات.

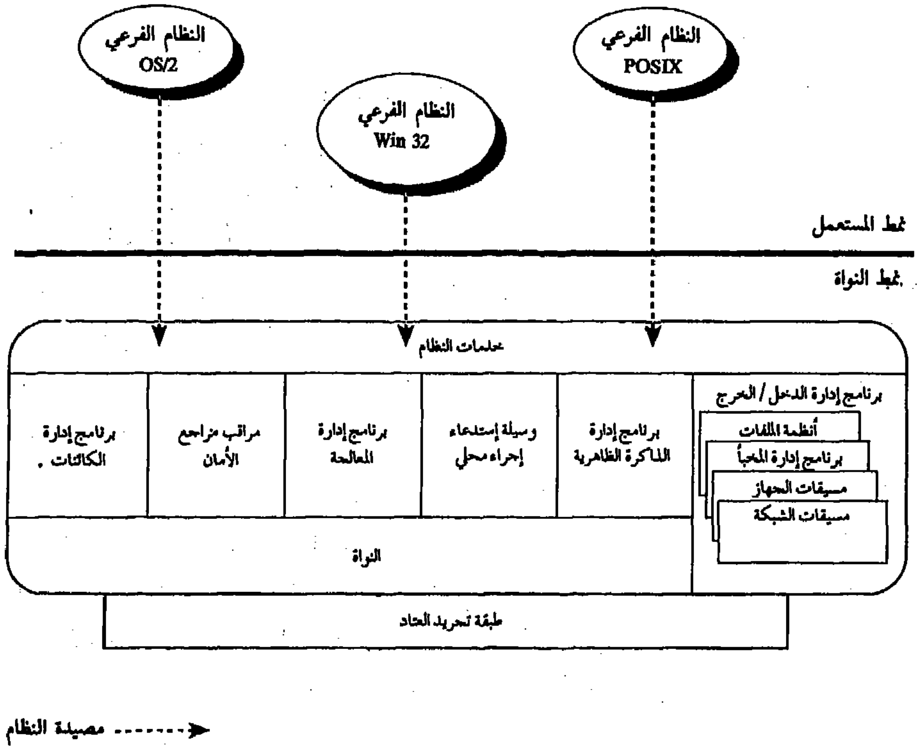
3-3-2-2 الخدمات المحليّة:

تطبّق الأنظمة الفرعيّة للمحيط روتينات API العائدة لها عن طريق إستدعاء الخدمات المحليّة NT، وهي خدمات النظام المتوفّرة من قِبَل المكوّنات الإفراديّة للبرنامج التنفيذي NT. يزوّد برنامج إدارة VM خدمات تخصيص الذاكرة وإلغاء تخصيصها، بينما يوفّر برنامج إدارة المعالجة خدمات إنشاء المعالجات والشعَب وإنهائها. وكما يوضّح الشكل (2-11)، وعندما يستدعي نظام فرعي خدمة محليّة NT، يكشف العتاد الإستدعاء وينقل التحكُّم إلى البرنامج التنفيذي NT. بعد ذلك تشتغل الخدمة في نمط النواة.

ولأن الخدمات المحليّة مستعملة من قِبَل عدّة أنظمة فرعيّة للمحيط، فيجب أن تكون شاملة ـ وحتى أوليّة. ويجب أن تكون مرنة دون قيود داخلية غير ضرورية. ولا يجب أن تولّد مؤثرات جانبيّة قد تتعارض مع حاجات الأنظمة الفرعيّة للمحيط.

إحدى الطرق التي تكون فيها الخدمات المحليّة مرنة هي قدرتها على العمل على أي معالجة يحدّدها المستدعي. فالمستدعي يزوّد مقبضاً للمعالجة، والخدمة تعمل على تلك المعالجة. فمثلاً، يستطيع نظام فرعي إستدعاء خدمة محليّة لإنشاء شعبة أوتخصيص ذاكرة لإحدى معالجات المستضاف. وطبعاً لا تستطيع معظم المعالجات العاديّة تنفيذ مثل هذه العمليات على المعالجات الأخرى. تحتوي الأنظمة الفرعيّة على صفات وصول قوية تضمن لها التحكّم بمستضافاتها.

الأنظمة الفرعيّة المحميّة و DLLs ومكوّنات البرنامج التنفيذي NT هي المستعملَين الأولين للخدمات المحليّة NT. وتكتب التطبيقات التي تشتغل على النظام Windows NT إلى أنظمة



الشكل (2-11)
استدعاء خدمة محليّة للنظام

تداخل البرمجة Wn 32 و MS-DOS و 16-bit Windows و POSIX و OS/2 المزوّدة من قبل الأنظمة الفرعيّة للمحيط.

4-3-2-2 الكائنات:

إن العديد، وربما جميع الخدمات المحليّة NT هي خدمات كائنيّة. وهذا يعني أنها تنفذ عملاً على كائن في البرنامج التنفيذي NT. تفتح شعبة مقبض إلى كائن ثم تستعمل هذا المقبض عند إستدعاء الخدمات لتعمل على الكائن.

تستخدم الموارد المشاركة بما فيها المعالجات والشعّب والملفات والذاكرة المشاركة ككائنات في البرنامج التنفيذي NT. وهذا يتيح لنظام التشغيل إستعمال أوجه التشابه ضمن الموارد ولاستعمال شيفرة عامة حيث أمكن لإدارتها. إن نظام الكائن NT هو نقطة التركيز لعدّة أنواع من مهام إدارة الموارد مثل تسمية الموارد ووضع الحدود (تسمى الحصص) على كميّة الموارد التي يمكن أن تستعملها كل معالجة، ومشاركة الموارد بين معالجتين وحماية الموارد ضدّ الوصول غير المسموح به.

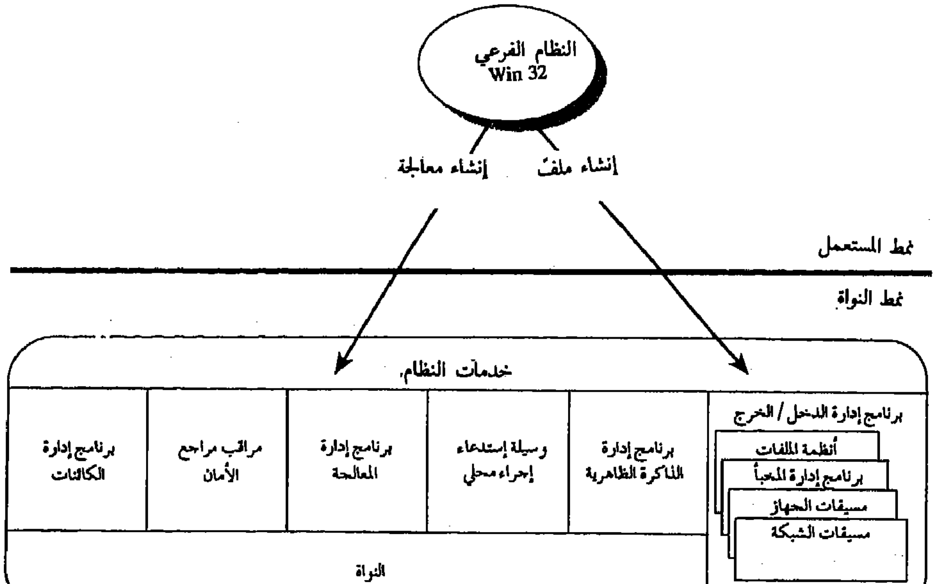
تستدعي الأنظمة الفرعيّة للمحيط بتكرار الخدمات الكائنيّة لإنشاء كائنات وفتح مقبض إليها ومناولتها أو حذفها. فمثلاً، إذا شغل المستعمل برنامجاً تطبيقياً Win 32 ـ مثلاً Microsoft Excel ـ يستدعي النظام الفرعي Win 32 برنامج إدارة المعالجة NT لإنشاء معالجة (المعالجة حيث سيشتغل البرنامج التطبيقي Excel) ويفتح مقبضاً إليه. ويستدعي برنامج إدارة المعالجة بدوره برنامج إدارة الكائن لإنشاء كائن معالجة وكائن شعبة. وبشكل مشابه، إذا حفظ المستعمل صفحة جدوليّة جديدة Excel، يستدعي النظام الفرعي Win 32 برنامج إدارة دخل / خرج NT لإنشاء كائن ملفّ يمثّل الملفّ حيث تخزّن الصفحة الجدوليّة وليفتح مقبضاً إلى الكائن. ويستدعي برنامج إدارة الدخل / الخرج برنامج إدارة الكائن للقيام بالوظيفة. يوضح ذلك الشكل (2-12) الموجود على الصفحة التالية.

تنفّذ معظم عمليات إدارة موارد NT عندما تنشىء بعض المعالجات كائناً و / أو تفتح مقبضاً إلى كائن. فمثلاً، عندما تنشىء معالجة (في هذه الحالة، النظام الفرعي Win 32) كائناً، فإنها تحدّد اختيارياً إسماً للكائن. وبتحديد إسم لكائن يجعل ذلك الكائن متوفراً للمشاركة من قِبَل المعالجات الأخرى. فالمعالجة التي تريد مشاركة الكائن تستردُّ إسم الكائن عن طريق إستدعاء برنامج إدارة الكائنات NT ثم تفتح مقبضاً إلى ذلك الكائن.

تحصّص الكائنات من ذاكرة نظام التشغيل. ولمنع أية معالجة من إستعمال الكثير من ذاكرة النظام، تحدّد حصة لكل المعالجات في كل مرة تفتح شعبها مقبضاً إلى نوع كائن معينّ. وإذا

إستهلكت المعالجة حصتها، لا يسمح برنامج إدارة الكائن لها بفتح مزيد من مقابض إلى الكائن.

إضافة إلى إدارة الموارد وتسهيل مشاركة الموارد، يستخدم نظام الكائن NT كنقطة تركيز لأمان الموارد. فعندما تفتح معالجة مقبض إلى كائن NT، ينشط النظام الفرعي للأمان NT. ويلحق بكل كائن قاعدة بيانات صغيرة، تسمى قائمة التحكُّم بالوصول (ACL) تحتـوي المعلومات المتعلّقة بالمعالجات التي تستطيع الوصول إلى الكائن وما يمكنها أن تقوم به. وعندما تفتح معالجة مقبض إلى كائن، فإنها تحدّد العمليّات التي تريد تنفيذها. فمثلاً، قد تفتح ملفاً للوصول إلى القراءة. يدقّق نظام الأمان لجهة أن هذه المعالجة مسموح لها الوصول إلى القراءة من ذلك الملف. وإذا إحتاج المستدعي للوصول إلى الكتابة إلى الملف فقد يطلب الوصول إلى القراءة وإلى الكتابة عندما يفتح المقبض الأول أو يستطيع فتح مقبض ثاني للوصول إلى الكتابة. ولأنه يتوجّب على المعالجة فتح مقبض إلى كائن قبل تمكّنها من القيام بأي شيء عليه ولأن فتح مقبض يحفز نظام الأمان، لا يمكن لأية معالجة تجاوز نظام الأمان NT.



الشكل (2-12)

إنشاء كائنات NT

5-3-2-2 الذاكرة الظاهريّة:

تعتمد أنظمة التشغيل معاينات مختلفة من الذاكرة الفعليّة وتطلب من برامجها الوصول إلى الذاكرة بطرُق محدّدة. وفي النظام Windows NT، تشتغل البرامج التطبيقيّة في محيط نظام تشغيل يتصرّف مثل النظام Windows وMS-DOS وPOSIX أو OS/2. أما التحدّي فيكمن في الإتاحة لكل أنواع التطبيقات الإشتغال دون إعادة كتابتها أو إصطدامها ببعضها البعض في الذاكرة.

توفّر كل الأنظمة الفرعيّة للمحيط في النظام Windows NT معاينة للذاكرة تتوافق مع ما تتوقّعه التطبيقات. وتحت الأنظمة الفرعيّة للمحيط، يحتوي البرنامج التنفيذي NT على بنية ذاكرة خاصة به، والتي توصل إليها الأنظمة الفرعيّة للمحيط عن طريق إستدعاء الخدمات المحليّة NT.

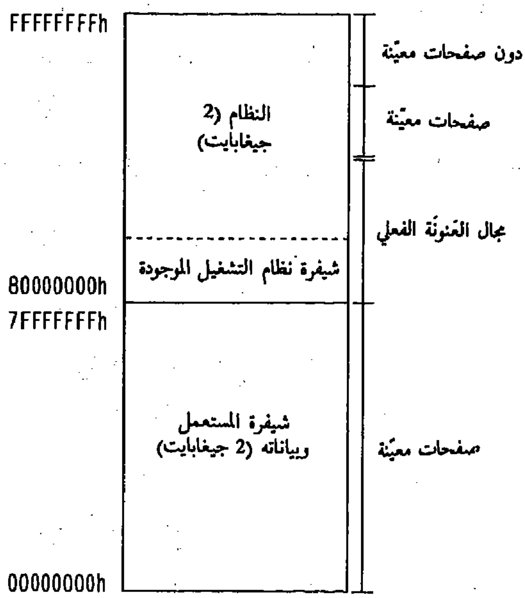
إن تصميم ذاكرة NT هو نظام ذاكرة ظاهريّة تعتمد على عناوين من 32 بتاً في فسحة عنوان مسطّحة (خطيّة). إن فسحة العنوان الظاهري للمعالجة هو مجموعة من العناوين المتوفّرة للإستعمال من قِبَل شعَب المعالجة. وخلال وقت التشغيل، يخطط أو يترجم برنامج إدارة VM بمساعدة العتاد، العناوين الظاهريّة إلى عناوين فعليّة حيث يتمّ تخزين البيانات. وبالتحكُّم بالتخطيط، يستطيع نظام التشغيل ضمان عدم إصطدام المعالجات الإفراديّة ببعضها البعض أو الكتابة فوق نظام التشغيل.

تبلغ فسحة العنوان الظاهري للمعالجة 4 جيغابايت (2^{32} بايت) مع حفظ 2 جيغابايت لتخزين البرنامج، وحفظ 2 جيغابايت لتخزين النظام. و4 جيغابايت (أو حتى 2) هي أكبر بكثير من كميّة الذاكرة الفعليّة المتوفّرة على الماكنات العاديّة. وعندما تمتلىء الذاكرة الفعليّة، ينقل برنامج إدارة VM أو يعيِّن صفحات لبعض محتويات الذاكرة على القرص. يخلي تعيين صفحات للبيانات على قرص الذاكرة الفعليّة بحيث يمكن إستعمالها لأشياء أخرى. يحمّل برنامج إدارة VM المعلومات إلى الذاكرة من القرص. يتمُّ وصف الذاكرة الظاهريّة بتفصيل أكبر في الفصل السادس، «برنامج إدارة الذاكرة الظاهريّة».

في النظام Windows NT، يستقرُّ نظام التشغيل في ذاكرة ظاهريّة عالية وتستقرُّ شيفرة المستعمل وبياناته في ذاكرة ظاهريّة مخفّضة كما يبيّن في الشكل (2-13) على الصفحة التالية. لا تستطيع شعبة في نمط المستعمل القراءة أو الكتابة إلى ذاكرة النظام مباشرة.

ولا يتمُّ أبداً تعيين صفحات لقسم من ذاكرة النظام، الذي يسمى المجمّع دون صفحات معيّنة، على قرص وهو يستعمل لتخزين بعض كائنات NT وبنيات البيانات المهمّة الأخرى.

ويتمّ تعيين صفحات لكل ذاكرة المستعمل. (راجع الفصل السادس «برنامج إدارة الذاكرة الظاهريّة»، للحصول على مزيد من المعلومات).



الشكل (2-13)

مخطّط تصميم فسحة العنوان NT

6-3-2-2 أنظمة الدخل / الخرج والملفّات:

كما هي الحال مع الذاكرة، توفّر الأنظمة الفرعيّة للمحيط وسائل للدخل / الخرج التي تتوقّعها التطبيقات. وهي تطبّق هذه الوسائل الإفراديّة بواسطة إستدعاء خدمات الدخل / الخرج NT المحليّة.

يستعمل نظام الدخل / الخرج المحلي نموذج دخل / خرج غير متزامن، لكنه يوفّر خدمات للنظام تتيح للأنظمة الفرعيّة للمحيط إستعمال إما نموذج متزامن أو غير متزامن. يتيح الدخل / الخرج غير المتزامن لمستدعي طلب عملية دخل / خرج ثم القيام بأعمال أخرى خلال إنهاء الجهاز نقل البيانات. ويبلغ نظام الدخل / الخرج المستدعي تلقائياً عند إتمام الدخل / الخرج لكي يقوم المستدعي بمعالجات لاحقة. ولأن أجهزة الدخل / الخرج أبطأ من المعالجات، فقد

يستطيع برنامج يقوم بالكثير من أعمال الدخل / الخرج، تحسين أدائها باستعمال الدخل / الخرج المتزامن.

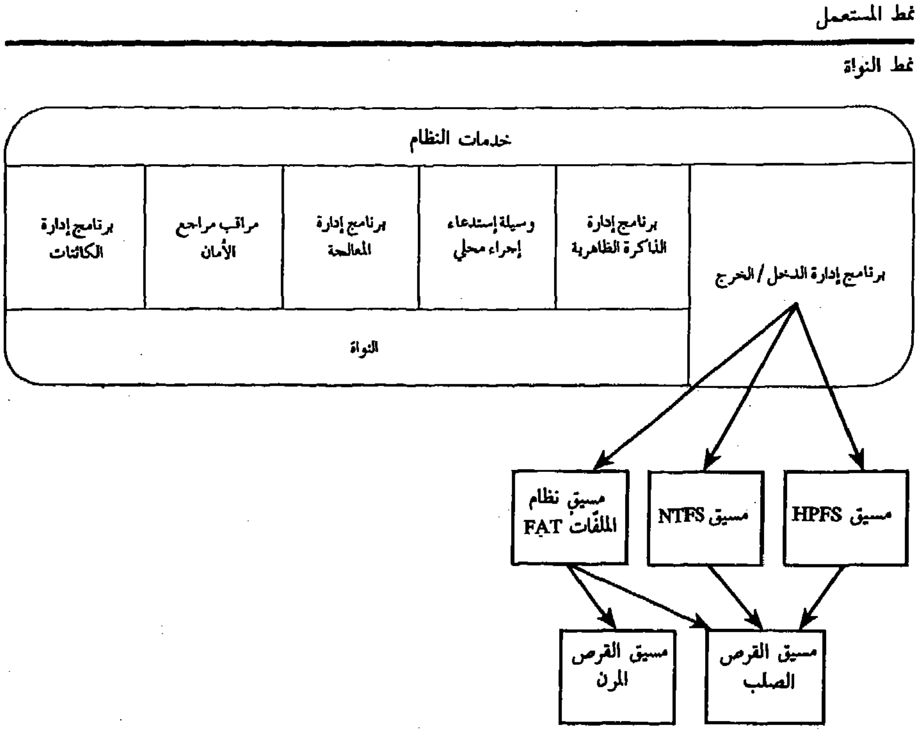
يدعم النظام Windows NT عدّة أنظمة ملفّات، بما فيها نظام ملفّات جدول تخصيص الملفّات (FAT) ونظام الملفّات المرتفع الأداء (HPFS) ونظام ملفّات جديد يسمّى نظام ملفّات NT (NTFS). يمدّد النظام NTFS القدرات الموجودة في نظامي الملفّات FAT و HPFS لإضافة المزايا الجديدة التالية:

- إستعادة نظام الملفّات الذي يتيح الإستعادة السريعة لبيانات قرصيّة بعد تعطُّل نظام.
- القدرة على مناولة وسيط تخزين كبير ـ بحجم حتى 2^{64} بايت، أو حوالي 17 مليار جيغابايت.
- مزايا أمان بما فيها ملفّات التنفيذ فقط.
- أسماء ملفّات أحادية الشيفرة، تتيح نقل المستندات من حاسوب إلى آخر عالمياً دون تشويه أسماء الملفّات وأسماء المسارات. (راجع القسم 2-3-1).
- دعم محيط نظام التشغيل POSIX، بما فيها الروابط الصلبة والأسماء الحاليّة والمعلومات المتعلّقة بآخر مرّة تمّ فيها فتح ملفّ.
- مزايا مدوديّة مستقبليّة، مثل العمليّات التبادليّة لدعم التطبيقات التي تسمح بالأعطال، وإعداد الملفّات للإصدار المحكوم من المستعمل ومجاري البيانات المتعدّدة لكل ملفّ والخيارات المرنة لتسمية الملفّات وصفات الملفّات ودعم ملقّمات الملفّات العامة الإستعمال.

يتيح برنامج إدارة الدخل / الخرج تحميل مسيقات الجهاز وأنظمة الملفّات (التي يعتبرها مسيقات «جهاز») دينامياً إلى النظام ومنه وفقاً لحاجات المستعمل. والمسيقات هي منظومة يمكن ترتيبها في طبقات واحدة على الأخرى والتي تتيح لأنظمة ملفّات مختلفة إستدعاء نفس مسيق القرص المرن أو مسيق القرص الصلب للوصول إلى الملفّات، كما يُبيِّن في الشكل (2-14).

كذلك يوفّر نموذج المسيق المرتّب بطبقات القدرة على إدراج مسيقات إضافيّة في التسلسل الهرمي. فمثلاً، تستطيع مسيقات نظام الملفّات المنطقي أو مسيقات الساحة بالأعطال إستيعاب المستويات المتوسطة في التسلسل الهرمي للمسيق.

يوفّر النظام Windows NT الوصول إلى الملفّات على شبكة برنامج الإدارة LAN Manager عبر مسيق نظام ملفّات يسمى الموجّه الجديد للنظام Windows NT . يقبل الموجّه الجديد الطلبات من الملفّات البعيدة ويوجّهها إلى ملقّم LAN Manager على ماكنة أخرى.



الشكل (2-14)
المسيقات المرتّبة في طبقات

3-2 التصاميم الإضافيّة للنظام Windows NT

لم تشمل هذه الجولة السريعة إلى الآن كل العناصر المهمّة في النظام Windows NT. غير أن شمل العديد من المواضيع سيؤجّل إلى آخر هذا الكتاب وستُترك المواضيع الأخرى للكتب اللاحقة. لكن يوجد موضوعان لا يتلاءمان بشكل مناسب في أي مكوّن من نظام التشغيل (أو أي فصل في هذا الكتاب) ومهمّة جداً بحيث لا يمكن حذفها. الموضوع الأول هو دعم التدويل للنظام Windows NT الذي يتيح للمستعملين في العديد من الدول التفاعل مع النظام في لغتهم الأُمّ. وهو يوفّر لمطوّري البرنامج التطبيقي الأدوات المطلوبة لكتابة البرامج التطبيقيّة العالميّة. والموضوع الثاني هو المناولة الإستثنائيّة البنيويّة، وهي مزيّة مزوّدة في Microsoft C ومدعومة بواسطة نواة NT. وهي تتيح للمستعملين كتابة البرامج التطبيقيّة القويّة. كذلك يستعمل النظام Windows NT المكتوب في معظمه باللغة C من Microsoft، مزايا المناولة الإستثنائيّة البنيويّة لجعل نظام التشغيل إعتمادي.

ولا يمكن عرض هذه المواضيع بشكل مناسب في صفحات قصيرة قليلة. لكن القسمين التاليين يوفّران لمحة سريعة عن ـ المواضيع المتعلّقة بالتدويل والمناولة الإستثنائيّة البنيويّة ويلخّصان كيفيّة عنونتها في النظام Windows NT. راجع المراجع لمعرفة مصادر المعلومات الإضافيّة.

1-3-2 التدويل:

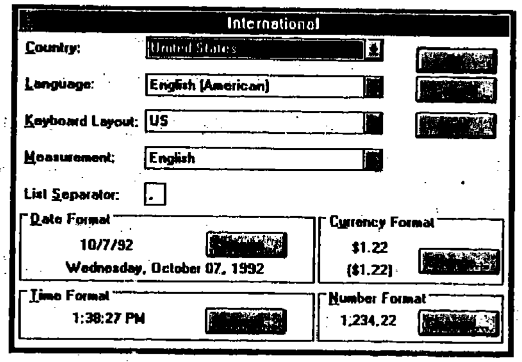
مع توفّر الرحلات النفّاثة والإتصالات البعيدة المعقّدة، أصبح العالم مكاناً أصغر. وبالتالي، أصبحت الأسواق العالميّة أكثر أهميّة لصناعة الحواسيب. وتساهم المبيعات العالميّة بحصّة كبيرة في سوق البرامج التطبيقيّة. إن هدف النظام Windows NT هو في جعله نظام تشغيل متعدّد اللغات وواحد يوفّر أُسس صلبة لتطوير البرامج التطبيقيّة العالميّة واستعمالها.

تظهر مظاهر الدعم الدولي للمستعمل في لوحة الحكم في Win 32 المبيّنة في الشكل (15-2).

لم تتغيّر خانة الحوار هذه من Windows 3.0. لكن يظلّ التداخل مع المستعمل، تغيّر الكثير. فالدعم الدولي واضح في النظام Windows NT، حيث يوفّر برامج خدماتيّة منظوميّة للتطبيقات ولمكوّنات النظام الهامّة مثل النظام الفرعي Win 32. وسيستمرّ تطوير التداخل مع المستعمل لتدويل البرامج الخدماتيّة في الإصدارات المستقبليّة.

1-1-3-2 الأسواق المحليّة:

تتّصف الأسواق المحليّة المختلفة بمتطلّبات مختلفة للبراجيّات. أهم هذه المتطلّبات هي



الشكل (15-2)
خانة الحوار دول ومقاييس **International**

الإتاحة للمستعمل التفاعل مع البراجيّات في لغته الأمّ باستعمال مصطلحات محليّة لعرض البيانات.

فالسوق المحلّي يتألف من لغة وبلد ومجموعة شيفرات تستعملها الشيفرات الثنائيّة لعرض أحرف لغة معيّنة (إن Windows ANSI هي إحدى مجموعة الشيفرات هذه). وعند تركيب النظام Windows NT، ينتقي المستعمل لغة ليستعملها ويعيّن لها سوقاً محلّياً مفترضاً. ويوفّر السوق المحلّي المفترض للمستعمل المفترضات الصحيحة لتصميم لوحة المفاتيح حيث يفرز الترتُّث والعملة ونسق التاريخ والوقت ويستطيع المستعمل تجاوز هذه المفترضات.

وفي الدول التي تتكلّم بعدّة لغات مثل كندا وسويسرا وبلجيكا، يحتاج المستعمل للقدرة على التحويل بين لغتين أو أكثر على أساس منتظم. لكن بعض الشركات ومن بينها Microsoft، تحتوي على فروع حيث تستعمل عدّة لغات مختلفة. ويجب أن يتمكّن كل مستعمل من التحويل بين الأسواق المحليّة في أي وقت لإرسال البيانات بين الأسواق المحليّة دون فقدان المعلومات. ولتحقيق ذلك، يجب فصل البرامج التطبيقيّة (وفي هذه الحالة، Windows) إلى قسمين:

- الشيفرة التي يمكن إستعمالها في كل الأسواق المحليّة.
- البيانات التي يجب ترجمتها للأسواق المحليّة المختلفة.

في Windows، تتألف فئة البيانات من موارد مثل القوائم والرسائل. تفصل هذه الموارد عن الجسم الرئيسي للشيفرة ويمكن إلحاقها أو فصلها عن Windows. وعندما يحوّل المستعمل السوق المحلّي، تتغيّر مجموعة الموارد لتمثّل السوق المحلّي الجديد. وبما أن مجموعة موارد Windows أصغر بكثير من النظام Windows نفسه، يمكن تحميل عدّة مجموعات موارد مختلفة خلال التركيب، حيث يُتاح للمستعمل التحويل بين الأسواق المحليّة المختلفة بسهولة دون تحميل ملفّات جديدة من الأقراص المرنة. إضافة لذلك، يمكن إرسال رزمة Windows NT واحدة إلى كل الدول ذات دعم محلّي مركّب بالداخل. أما المهمّة الوحيدة المتبقية فهي في ترجمة ملفّات الموارد والمستندات.

لتسهيل المركّزة المحليّة، يوفّر النظام الفرعي Win 32 في Windows NT روتينات API لدعم اللغات المحليّة (NLS) توفّر للتطبيقات الوصول إلى مقارنات التنضيد الصحيحة، وجداول ترتيب فرز أحرف اللغات المختلفة وروتينات نسق التاريخ والوقت والعملة وروتينات تحديد السوق المحلّي المنتقى والأسواق المحليّة الأخرى الموجودة على النظام. إضافة لذلك، توفّر روتينات NLS API الروتينات للتحويل بين مجموعة الشيفرة الدوليّة المستعملة من قِبَل النظام Windows NT ومجموعات الشيفرة المستعملة عموماً. (يُشرح هذا الموضوع بتفصيل أكبر في

القسم التالي). ويوفّر النظام الفرعي WIN 32 ومكتبة وقت التشغيل C روتينات API خاصة بها تعتمد على NLS. ويُتيح إستعمال هذه البرامج الخدماتيّة للتطبيقات دعم المركّزة المحليّة دون الحاجة لإستنساخ قاعدة البيانات الأساسيّة (الجداول ومجموعات الشيفرة وما شابه) المطلوبة للقيام بذلك.

2-1-3-2 الشيفرة الأحادية:

الطبقة الأسفل لدعم المركّزة المحليّة هي في عرض الأحرف الإفراديّة، مجموعات الشيفرة. لقد إستخدمت الولايات المتحدة الأميركية نظام ASCII (القانون القياسي الأميركي لتبادل البيانات) لعرض البيانات. وبالنسبة للأوروبيين والدول الأخرى، لا يناسب النظام ASCII لأنه يفقد الرموز وعلامات الترقيم العامة. فمثلاً، تُحذف علامة الجنيه الإسترليني وكذلك العلامات اللهجيّة في اللغة الفرنسيّة، والألمانيّة والهولنديّة والإسبانيّة.

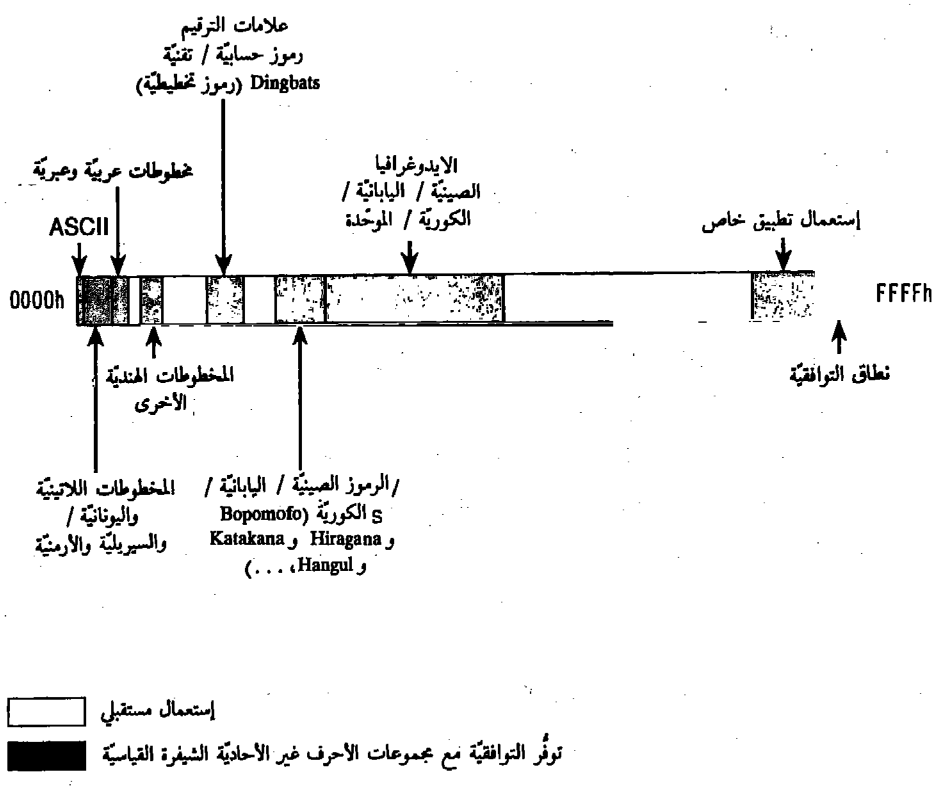
أنشأت منظمة المواصفات القياسيّة الدوليّة (ISO) مجموعة شيفرة قياسيّة، سُمِّيَت Latin 1 (مواصفات ISO رقم8859-1) والتي تحدّد شيفرات كل الأحرف الأوروبيّة المحذوفة من قبل ASCII. يستعمل النظام Windows من Microsoft تعديلاً بسيطاً على Latin 1 ويسمى مجموعة شيفرات ANSI للنظام Windows. ومجموعة Windows ANSI هي مخطّط تشفير من بايت واحد لأنه يستعمل 8 بتاً لعرض كل حرف والعدد الأقصى للأحرف التي يمكن عرضها باستعمال 8 بتاً هو 256 (2^8).

المخطوطة هي مجموعة من الأحرف المطلوبة للكتابة في لغة معيّنة. وتستعمل نفس المخطوطة غالباً لعدّة لغات. (فمثلاً، تستعمل المخطوطة Cyrillic للغتين الروسية والأوكرانيّة). يستطيع مخطّط Windows ANSI ومخطّطات التشفير الأخرى الأحاديّة البايت تشفير ما يمكن من الأحرف لعرض أحرف المخطوطات الغربيّة. لكن يتعذّر تشفير المخطوطات الشرقيّة كاليابانيّة والصينيّة، والتي تستخدم الآلاف من الأحرف المستقلّة، باستعمال مخطّط تشفير أحادي البايت. وتُخزَّن هذه المخطوطات عادة باستعمال مخطّط تشفير مزدوج البايت والذي يستعمل 16 بتاً لكل حرف أو مخطط تشفير متعدّد البايتات، حيث تعرض بعض الأحرف بتتابع 8 بتاً وتعرض أخرى بتتابع من 16 بتاً و24 بتاً أو32 بتاً. ويحتاج هذا المخطّط الأخير إلى لوغارتيم قوي للتحليل النحوي وذلك لتحديد عرض التخزين لحرف معيّن. إضافة لذلك، فإن تكاثر مجموعات شيفرة مختلفة يعني أن كل شيفرة معيّنة قد تؤدّي إلى أحرف مختلفة كلياً على حاسوبين مختلفين، وفقاً لمجموعة الشيفرة التي يستعملها كل حاسوب.

كل مشكلة مخطّطات التشفير المتعدّدة، ولاستيعاب مجموعة مخطوطات أكثر شموليّة،

يستخدم النظام Windows NT المواصفات الجديدة الشيفرة الأحاديّة Unicode لعرض البيانات. تستطيع الشيفرة الأحاديّة Unicode، وهي مخطّط تشفير الأحرف من 16 بتاً، عرض 65,536 (2^{16}) حرفاً. وهذا العدد يكفي ليشمل كل اللغات في سوق الحواسيب حالياً وكذلك العديد من اللغات القديمة والسريّة مع تطبيقات محدودة (مثل السنسكريتية، وبالتالي الهيروغليفيّة الفرعونيّة). كذلك تشمل الشيفرة الأحاديّة عروض لعلامات الترقيم والرموز الحسابيّة ومجموعة من الحروف التخطيطيّة التي تسمى dingbats، مع توفّر فسحة كافية للتوسع المستقبلي.

تفصل الشيفرة الأحاديّة جوهر الحرف من الخط وتنسق المعلومات المستعملة لعرضه. وتوافق كل شيفرة حرفاً واحداً فقط، وتطبّق معلومات الخط على أحرف الشيفرة الأحاديّة لتعرضها في أشكال مختلفة. يوضح الشكل (2-16) تصميم المخطوطات والرموز في الشيفرة الأحاديّة Unicode.



الشكل (2-16)

تصميم الشيفرة الأحاديّة Unicode

رغم أن النظام الفرعي Win 32 يوفّر روتينات API لتنضيد ANSI و Unicode، غير أن Unicode هي مجموعة شيفرة محليّة في النظام Windows NT. وتعرض كل نضائد الأحرف في النظام، بما فيها أسماء الكائنات وأسماء المسارات وأسماء الملفّات وأسماء الأدلّة، أحرف Unicode من 16 بتاً. حتى أن النظام الفرعي Win 32 يحوّل أيّة أحرف ANSI يستلمها إلى نضائد Unicode قبل مناولتها ثم يحوّلها إلى ANSI، عند الضرورة، عند الخروج من النظام.

إن إستعمال Unicode يُزيل كل التقييدات على مجموعة الأحرف التي يستطيع النظام Windows NT عرضها. ولأن الشيفرة Unicode تحقّق شيفرة فريدة لكل حرف من كل مخطوطة، يستطيع النظام Windows NT ضمان دقة ترجمة الأحرف المتبادلة من النظام وإليه دائماً.

2-3-2 المناولة الإستثنائيّة البنيويّة:

يسمى التصميم الثاني الخاص المدعوم والمستعمل من قِبَل النظام Windows NT بالمناولة الإستثنائيّة البنيويّة. الإستثناءات هي أخطاء متزامنة أو أحداث تؤدّي إلى تنفيذ شيفرة خارج إنسياب التحكّم العادي. وبعكس المقاطعات، التي تولّد من مصدر خارجي، تحصل الإستثناءات عندما ينفّذ برنامج تتابع شيفرة معيّنة ويعاود توليد الإستثناءات.

فمثلاً، عندما يستدعي برنامج وظيفةC malloc ()، وتكون النتيجة عادة في قيام الوظيفة malloc () بتحصيص ذاكرة وإرجاع مؤشّر إليه. تحصل الحالة الإستثنائيّة عندما تسبّب بعض المشاكل، مثل عدم توفّر ذاكرة، في إخفاق التحصيص. في هذه الحالة، ترجع الوظيفة مؤشّر NULL.

إن إرجاع قيمة خاصة تشير إلى أن الإستثناء هو شكل عام لكن أوّلي للمناولة الإستثنائيّة، وأنه يتّصف ببعض التردّدات. فأولاً، يجب على المبرمج أن يدقّق بإمعان بالقيمة الرجعيّة ويصلّح الأخطاء أو يُرسلها إلى طبقة برمجيّات أعلى. فإذا حُذفت طبقة واحدة التدقيق، تظهر العِلل في أجزاء غير متعلّقة من البرنامج. وثانياً، تصبح الشيفرة مملوءة بالفقرات If ... Then... Else التي تتناول الحالة الشاذّة عِوَضاً عن الحالة النموذجيّة. ثالثاً، قد لا تتوفّر بسهولة المعلومات المتعلّقة بسبب تعطّل العمليّة إلى الشيفرة التي يجب أن تعالج المشكلة.

يمكن كشف الإستثناءات إما بواسطة العتاد أو بالبراجيّات. فمثلاً، تكشف العتاد عموماً إستثناءات القسمة على صفر، بينما تكشف البراجيّات مخالفات الوصول إلى الذاكرة. إن المناولة الإستثنائيّة البنيويّة هي الطريقة المستعملة في النظام Windows NT لمعالجة إستثناءات العتاد والبراجيّات، باستعمال بنية التحكّم (وبالتالي الإسم) للغة البرمجة. تتيح المناولة الإستثنائيّة

البنيويّة لأية كتلة شيفرات تحديد نوع أو أنواع الشيفرات التي تريد حمايتها وتسجيل تتابع شيفرة خاص (مناول الإستثناءات) الذي ينفّذ إذا حصلت مثل هذه الإستثناءات ضمن كتلة الشيفرات المحميّة.

إن الشيفرة هي روتين بسيط مكتوب بلغة C من Microsoft يتضمّن مناول إستثناءات. وهو إصدار معدَّل لوظيفة مكتبة C القياسيّة () strlen التي ترجع طول التنضيد المنتهي بصفر.

```
/* safelen: return valid length of string s,
   even if the string pointer was bad */

int safelen(char *s)
{
    int count = 0;

    try {
        while (*s++ != '\0')
        /* possible access violation */
           count++;
        return (count);
        }
    except (GetExceptionCode() == ACCESS_VIOLATION ?
                          EXCEPTION_EXECUTE_HANDLER :
                          EXCEPTION_CONTINUE_SEARCH)
        {
    /*  pointer was bad or string was not terminated */
        return (count);
        }
}
```

تتمهّل الوظيفة العاديّة () strlen في الذاكرة بمعدّل حرف واحد في كل مرة إلى أن تجد حرف NULL. لكن إذا كان التنضيد غير منتهي بصفر أو كان مؤشّر التنضيد غير صالح، تنتهي الوظيفة () strlen بشكل غير متوقّع مع إستثناء مخالفة في الوصول.

يلتقط هذا الإصدار المعدّل من الشيفرة الإستثناء ويرجع قيمة صالحة (وليس بالضرورة قيمة صحيحة، بل بالكاد صالحة) عِوَضاً عن إنهاء البرنامج. تستعمل الكلمة المرجعيّة الجديدة في اللغة C، try، لتعليم بداية كتلة الشيفرة وينقل التحكّم إلى الكلمة المرجعيّة except التي تتبع (ضمن أقواس) بمرشح إستثناء. يتيح مرشح الإستثناء للمبرمج تحديد تنفيذ مناول الإستثناءات فقط على أنواع الإستثناءات المنتقاة. وإذا انتقل مرشح الإستثناء إلى TRUE، عندها ينفذ مناول الإستثناءات ــ في هذه الحالة، الإيعاز الرجيع (count). إن مراشح الإستثناءات قوية لأنها

تستطيع الوصول إلى البيانات المحليّة وأن تكون بتعقيد تحكُّمي. وهي تتيح تنفيذ مناول الإستثناءات بظل حالات دقيقة. تسمى عملية نقل التحكُّم إلى مناول إستثناءات بعملية رفع إستثناء. لاحظ كيف يتمُّ إزالة شيفرة مناولة الخطأ من الخط الرئيسي للبرنامج.

يمكن لكل كتلة شيفرة أن تحتوي مناول إستثناءات مستقلا ويمكن تعشيش مناولات الإستثناءات ضمن بعضها البعض. وعند حصول إستثناء، يستطيع مرشّح الإستثناء إختبار نوع الإستثناء والطلب من نظام التشغيل تنفيذ مناول الإستثناءات ومواصلة البرنامج أو إنهاء البرنامج أو البحث عن مناول إستثناءات في كتلة شيفرة محصورة.

إستثناءات نظام التشغيل ليست الإستثناءات الوحيدة التي يستجيب لها نظام التشغيل. فالبرامج التطبيقيّة تستطيع توليد إستثناء بإستعمال روتين RaiseException () في Win 32 API، حيث تنقل التحكُّم إلى مناول إستثناءات مسجّل. يدعم نظام التشغيل هذه العمليّة عن طريق تسجيل مناولات الإستثناءات والبحث عنها بالترتيب الصحيح عند رفع الإستثناءات. وإذا لم يعالج أي مناول إستثناءات المشكلة، ينهي نظام التشغيل البرنامج الذي سبّب الخطأ. إن وسيلة مناولة الإستثناءات في النظام Windows NT ليس خاصاً بلغة. تستعمل آلية واحدة عبر كل اللغات. وتعرّف كل لغة كيفيّة كشف آلية المناولة الإستثنائيّة الأساسيّة.

يتيح نوع آخر من مناول الإستثناءات والمعروف بإسم مناول الإنهاء، لبرنامج تطبيقي ضمان تنفيذ كتلة شيفرة معيّنة دائماً حتى إذا انتهت كتلة شيفرة محميّة بطريقة غير متوقّعة. تحتوي مناولات الإنهاء في غالب الأحيان شيفرة تخلي الموارد المحصصة بحيث إذا إنتهى إجراء بشكل غير متوقّع، تفلت الموارد المحصّصة إلى النظام. إن التالي هو جزء شيفرة WN 32 يوضح الغرض من مناول الإنهاء:

القسم الحرج هو كائن مزامنة Win 32 يضمن تنفيذ شعبة واحدة فقط لكتلة شيفرة معيّنة في كل مرة. في هذا المثال، تتمكّن الشعبة من الوصول إلى القسم الحرج وتحصّص مخزناً مؤقّتاً ثم تعدّل المخزن المؤقّت. وإذا حصل خطأ ما (ربما إستثناء غير مناول) وجعل الروتين ينتهي خلال وجود الشعبة في القسم الحرج، يتمُّ منع أية شعبة أخرى تنتظر الحصول على الموارد. إضافة لذلك، يفقد المخزن المؤقّت الذي حدّدته الشعبة ولا يتمكّن نظام التشغيل من إستعادته. (يسمي المطوّرون هذه الأنواع من الأخطاء بتسرُّبات الذاكرة. وإذا حصل الكثير منها، تصرف الذاكرة المتوفرة تدريجياً). يضمن مناول الإنهاء قيام الشعبة بإفلات كائن القسم الحرج وإخلاء المخزن المؤقت. تنفذ دائماً مناولات الإنهاء عندما يغادر إنسياب التحكُّم جسم الكتلة try... Finally دون إعتبار لكيفيّة الخروج.

```
/* allocate and initialize a
   global critical section object */
      .
      .
      .
LPSTR Buffer;
Buffer = NULL;

/* enter the critical section and
   allocate a buffer */
try {
      EnterCriticalSection(&CriticalSection)
      Buffer = LocalAlloc(LMEM_FIXED, 10);
      if(!Buffer) {
          return;
      }
      strcpy(Buffer, "Hello");
}
finally {
/* always leave the critical section and
   free the allocated buffer */
      if(Buffer != NULL)
          LocalFree(Buffer);
      LeaveCriticalSection(&CriticalSection);
 }
      .
      .
      .
```

يمكن إستعمال مناولات الإستثناء ومناولات الإنهاء بشكل مستقلّ أو بتوليفة لتحقيق الأداء القوي في أي برنامج تطبيقي. يستعمل النظام Windows NT كلاهما لضمان الأداء القوي في كل مستويات النظام.

4-2 في الختام:

هذه هي بعض ميزات النظام Windows NT. وهي قاعدة متناظرة لنظام تشغيل متعدّد المعالجة تدعم محيطات نظام التشغيل المتعدّدة. ويحتوي النظام Windows NT على تداخل تخطيطي مع المستعمل Windows ويشغّل برامج Win 32 و 16-bit Windows و MS-DOS و POSIX و OS/2. وهو يستخدم مبادىء متقدّمة لنظام التشغيل مثل الذاكرة الظاهريّة والمهام المتعدّدة الوقائيّة والمناولة الإستثنائيّة البنيويّة وكائنات نظام التشغيل. وهو نظام آمن وقوي واعتمادي ومرن. وهو يتّصف بقدرات كتلك التي كانت موجودة في أنظمة تشغيل الحواسيب

الصغيرة والكبيرة. بمعنى آخر، إن النظام Windows NT هو قطار سريع محجّم إلى رزمة بحجم لوح التزحلق ــ وقد تمثّل مستقبل الحوسبة على سطح مكتب. إن الفصول التالية في هذا الكتاب تشرح تفاصيل النظام Windows NT بدءاً من الكائنات ــ ووسائله في عرض الموارد وإدارتها وحمايتها.

3

برنامج إدارة الكائنات وأمان الكائنات

لقد أصبحت اللغات الكائنيّة وأنظمة التداخل مع المستعمل وأنظمة التشغيل من أهمّ المواضيع المتداولة في النصف الثاني من الثمانينات. وقد أعلنت الكائنات على أنها العلاج لكل مرض برمجة. لكن الكائنات ليست جديدة. فقد ظهرت لأول مرة في أواخر الستينات في لغات البرمجة مثل Simula التي طوّرت لإنشاء برامج محاكاة. وقد حاكت محاكات الحاسوب تصرُّف الكائنات الفعليّة. لذلك فإن البرمجة الكائنيّة التي توفّر طريقة لعرض ومناولة الكائنات الفعليّة والمجرّدة هي الطريقة الطبيعيّة في ذلك المجال.

كذلك تتناول أنظمة التشغيل الكائنات. وتتّخذ كائناتها شكل موارد العتاد مثل أجهزة الدخل / الخرج والذاكرة أو موارد البراجيّات مثل الملفّات والمعالجات والإعلام الإشاري. وتركّز معظم أنظمة التشغيل على الإختلافات بين هذه الموارد المتشاركة وتتناول نوعاً من الموارد بطريقة مختلفة، لكن إستعمالها ككائنات يستغلّ أوجه الشبه فيها. وهي تركّز كل إدارات الموارد في موقع واحد وتوفّر نموذجاً متماسكاً لإستعمال الموارد.

تبدأ الجولة في النظام Windows NT من البرنامج التنفيذي NT وبالتحديد من كائنات البرنامج التنفيذي NT. ومن الصعب البدء في أي مكان آخر لأن المعالجات والشعَب والملفّات وحتى النظام الفرعي Win 32 (معالجة) هي كائنات. وهكذا، فإن استيعاب نظام الكائن NT يوفّر السبيل إلى الأجزاء الواسعة من نظام التشغيل.

يشرح القسم الأول من هذا الفصل أنواع الكائنات الموجودة في النظام Windows NT ويصف كيفيّة إستعمالها. إن شرح بنية الكائن وكيفيّة إدارة برنامج إدارة الكائنات للكائنات هو موضوع القسم الثاني. ويركّز القسم الثالث على المهام الأساسيّة لنظام الأمان في Windows NT وهي حماية الكائنات.

1-3 كائنات البرنامج التنفيذي NT

ما هو الكائن؟ في البرنامج التنفيذي NT، الكائن هو لحظة وقت تشغيل واحدة لنوع كائن محدّد إستثنائياً. يتألف نوع كائن (يسمّى أحياناً فئة كائن) من نوع بيانات معرّفة بالنظام والخدمات التي تعمل على لحظات نوع البيانات ومجموعة من صفات الكائن. فإذا كتبت تطبيقات WIN 32، فإنك تواجه كائنات معالجة وشعبة وملفّ وحدث، على سبيل المثال. وتعتمد هذه الكائنات على كائنات بمستوى أدنى تمّ إنشاءها وإدارتها بواسطة البرنامج التنفيذي NT. وفي NT، المعالجة هي لحظة نوع كائن المعالجة والملفّ هو لحظة نوع كائن الملفّ وهكذا.

صفة الكائن هي مجال بيانات في كائن يعرّف جزئياً حالة الكائن. فكائن من نوع التكديس، على سبيل المثال، يحتوي مؤشّر تكديس كأحد أهم صفاته. تقرأ عادة خدمات الكائن، التي هي وسائل لمناولة الكائنات، صفات الكائن أو تغيّرها. فمثلاً، تغيّر خدمة الدفع لكائن تكديس قيمة مؤشّر التكديس.

إن الإختلاف الأساسي بين كائن وبنية بيانات عادية هو في كون البنية الداخليّة لكائن مخفيّة. ويجب أن تستدعي خدمة كائن لإخراج البيانات من كائن أو لوضع البيانات فيه. ولا تستطيع قراءة البيانات أو تغييرها مباشرة داخل كائن. وهذا الأمر يفصل التطبيق الأساسي للكائن عن الشيفرة التي تستعمله وهي طريقة تتيح تغيير تطبيق الكائن بسهولة.

لقد قرّر فريق تصميم البرنامج التنفيذي NT إستعمال الكائنات لعرض موارد النظام لأن الكائنات توفّر وسائل مركزيّة لتحقيق ثلاثة مهام أساسيّة لنظام التشغيل:

- توفير أسماء إنسان مقروءة لموارد النظام.
- مشاركة الموارد والبيانات ضمن المعالجات.
- حماية الموارد من الوصول غير المسموح به.

ليست كل بنيات البيانات في البرنامج التنفيذي NT كائنات. توضع في كائنات فقط البيانات التي يجب مشاركتها وحمايتها وتسميتها أو جعلها مرئيّة للبرامج في نمط المستعمل (عبر خدمات النظام). والبنيات المستعملة من قبل مكوّن برنامج تنفيذي واحد لتطبيق الوظائف الداخلية، مثلاً، وليست كائنات.

رغم إستعمال الكائنات لعرض موارد النظام المتشاركة، فالنظام Windows NT ليس نظاماً كائنياً كلياً. فمعظم شيفرات نظام التشغيل مكتوبة باللغة C للنقليّة. ويسبب توفّر أدوات

التطوير، لكن اللغة C لا تدعم مباشرة الإنشاءات الكائنيّة مثل الربط الدينامي لأنواع البيانات والوظائف المتعدّدة الأشكال أو تأصُّل الفئات. لذلك تستعير التطبيقات التي تعتمد على اللغة C في النظام Windows NT للكائنات، من المزايا الخاصة بلغات كائنيّة معيّنة، ولكن لا تعتمد عليها.

برنامج إدارة الكائنات هو مكوّن البرنامج التنفيذي NT المسؤول عن إنشاء كائنات NT وحذفها وحمايتها وتعقّبها، يمركز برنامج إدارة الكائنات عمليات التحكّم بالموارد التي يمكن أن تنتشر في خلاف ذلك عبر نظام التشغيل. لقد صمّم كل من Lou Perazzoli مدير الهندسة ورئيس مشروع تطوير النظام Windows NT و Steve Wood مبرمج قديم لأنظمة التشغيل من Microsoft، برنامج إدارة الكائنات ووضعوا أهداف الإستعمال التالية:

- توفير آلية عامة متناسقة لإستعمال موارد النظام.
- عزل حماية الكائن إلى موقع واحد في نظام التشغيل لتحقيق التوافق مع الأمان للحكومة الأميركيّة من الفئة C2.
- إنشاء مخطّط لتسمية الكائنات يستطيع شمل الكائنات الموجودة مثل الأجهزة والملفّات وأدلّة نظام ملفّات أو مجموعات مستقلّة أخرى من الكائنات.
- إنشاء طريقة تحدّد الكلفة لإستعمال الكائنات من قِبَل المعالجات بحيث يستطيع مدير النظام وضع حدود لإستعمال موارد النظام.
- إنشاء قواعد متناسقة لاحتجاز الكائن (أي، إبقاء الكائن متوفراً إلى أن تنتهي المعالجات من إستعماله).
- دعم متطلّبات محيطات أنظمة التشغيل المختلفة، مثل قدرة معالجة على تأهُّل الموارد من المعالـجة الأم (المطلوبة من قبل Windows و POSIX) والقدرة على إنشاء أسماء ملفّات حالاتيّة (مطلوبة من قِبَل POSIX).

تعرض الأقسام التالية أُسس كائنات البرنامج التنفيذي NT بما فيها كيفيّة بناء هذه الكائنات وكيفيّة إستعمالها في نظام التشغيل.

3-1-1 إستعمال الكائنات:

يستخدم البرنامج التنفيذي NT نوعين من الكائنات: كائنات تنفيذيّة وكائنات نواة. الكائنات التنفيذيّة هي الكائنات المستعملة من قِبَل المكوّنات المختلفة للبرنامج التنفيذي NT.

وهي متوفّرة لشيفرة نمط المستعمل (أنظمة فرعيّة محميّة) عبر خدمات NT المحليّة، ويمكن إنشاءها ومناولتها من قِبَل الأنظمة الفرعيّة أو من قِبَل البرنامج التنفيذي NT.

كائنات النواة هي مجموعة أوليّة من الكائنات المستعملة من قِبَل نواة NT، وهذه الكائنات غير مرئيّة لشيفرة نمط المستعمل لكنّها تنشأ وتستعمل فقط ضمن البرنامج التنفيذي NT. توفّر كائنات النواة قدرات أساسيّة مثل القدرة على تعديل جدولة النظام الممكن تحقيقها فقط بواسطة الطبقة الأدنى لنظام التشغيل ـ النواة. تحتوي العديد من كائنات النواة (تغلّف) كائن نواة واحد أو أكثر. لكن سيتمّ التركيز الآن على أنواع الكائنات المرئيّة للمستعمل المدرجة في الجدول (3-1) مع المكوّنات التنفيذيّة التي تعرّفها.

يعكس كل نظام فرعي للمحيط في Windows NT لتطبيقاته رسماً مختلفاً لنظام التشغيل. والكائنات التنفيذيّة والخدمات الكائنيّة هي مجموعة أوليّة تستعملها الأنظمة الفرعيّة للمحيط لإنشاء إصداراتها الخاصة عن الكائنات والموارد الأخرى. وقد تكون مجموعة الكائنات التي يزوّدها النظام الفرعي لمحيط إلى تطبيقاته أكبر أو أصغر من تلك التي يوفّرها البرنامج التنفيذي NT، ولا تدعم بعض الأنظمة الفرعيّة، مثل POSIX، الكائنات ككائنات بتاتاً. فالنظام الفرعي POSIX يستعمل الكائنات التنفيذيّة والخدمات كأساس لعرض المعالجات والأنابيب والموارد الأخرى من النوع POSIX إلى تطبيقاته. بينما تستعمل الأنظمة الفرعيّة الأخرى، مثل النظام الفرعي Win 32، كائنات البرنامج التنفيذي NT لإنشاء إصداراتها الخاصة من الكائنات. ويزوّد النظام الفرعي Win 32 إلى تطبيقات Win 32 الخوافت والإعلام الإشاري وكلاهما يعتمد مباشرة على كائنات البرنامج التنفيذي NT. إضافة لذلك، يزوّد النظام الفرعي Win 32 موارد الأنابيب والثقوب البريديّة المسمّاة التي تعتمد على كائنات ملفّ البرنامج التنفيذي NT.

يركّز هذا الفصل على الكائنات التنفيذيّة، تلك المتوفّرة من قِبَل البرنامج التنفيذي NT. ولا يجب الخلط بين الكائنات التنفيذيّة والكائنات المتوفّرة للبرامج التطبيقيّة بواسطة روتينات Win32 API و POSIX API أو OS/2 API.

1-1-1-3 النموذج الملفّي:

من ناحية البرمجة، يبدو النظام Windows NT مثل النظام Windows أو MS-DOS أو POSIX أو OS/2. ويرغب مبرمجو النظام الذين يكتبون نظاماً فرعياً لمحيط أو نظام ملفّات أو مسيّق لجهاز محلّي أو تطبيق خاص آخر، في معرفة المزيد حول الكائنات التنفيذيّة وإستعمالها مباشرة.

نوع الكائن التنفيذي	مُعرَّف من قِبَل	يمثّل
معالجة	برنامج إدارة المعالجة	إنفاذ برنامج، يشمل فسحة العنوان والموارد المطلوبة لتشغيل البرنامج
شعبة	برنامج إدارة المعالجة	كينونة قابلة للتنفيذ ضمن معالجة
قسم	برنامج إدارة الذاكرة	منطقة من الذاكرة المشاركة
ملف	برنامج إدارة الدخل / الخرج	حالة آنية للملفّ مفتوح أو جهاز دخل / خرج
منفّذ	برنامج خدماتي LPC	مقصد رسائل ممرّرة بين المعالجات
صفة وصول	نظام الأمان	بطاقة تعريف مقاومة للعبث تحتوي معلومات الأمان حول مستعمل مسجّل
حدث	خدمات دعم البرنامج التنفيذي	إعلان حصول حادثة في النظام
زوج حدث	خدمات دعم البرنامج التنفيذي	إعلان عن نسخ شعبة مستضاف مخصّصة لرسالة إلى ملقّم Win 32 أو بالعكس (مستعملة فقط من قِبَل النظام الفرعي Win 32).
إعلام إشاري	خدمات دعم البرنامج التنفيذي	عدّاد ينظّم عدد الشعَب التي تستطيع إستعمال الموارد
خوافت	خدمات دعم البرنامج التنفيذي	آلية توفّر قدرات عزل متبادلة لمحيطات Win 32 و OS/2
موقّت	خدمات دعم البرنامج التنفيذي	عدّاد يسجّل مرور الوقت
دليل الكائن	برنامج إدارة الكائنات	جهاز تنفّسي ذاكرتي لأسماء الكائنات
ربط رمزي	برنامج إدارة الكائنات	آلية للعودة بطريقة غير مباشرة إلى إسم كائن
إستمثال	النواة	آلية لقياس توزيع وقت التنفيذ ضمن كتلة شيفرة (لضبط الأداء)
مرجعي	برنامج إدارة التشكيل	فهرس للسجلّات في قاعدة بيانات تشكيل النظام Windows NT

تنشأ عادة الكائنات التنفيذيّة إما من قِبَل نظام فرعي محمي كإستجابة فوريّة لبعض نشاطات المستعمل أو من قبل المكوّنات المتعدّدة لنظام التشغيل كجزء من عملياته العاديّة. فمثلاً، لإنشاء ملفّ، يستدعي برنامج تطبيقي Win 32 روتين CreateFile () من Win 32 API. ويستدعي بدوره النظام الفرعي Win 32 خدمة NT محليّة تنشىء كائن ملفّ تنفيذي. وعندما يقرأ البرنامج التطبيقي الملفّ أو يكتبه لاحقاً، يستعمل النظام الفرعي Win 32 والبرنامج التنفيذي NT كائن الملفّ للوصول إلى الملفّ.

تمثّل عمليّات الملفّ حالة شاذّة في نظام الكائن NT لأن الملفّات هي موارد مثابرة ولا تتواجد في الذاكرة. لكن الملفّات هامة لأن النموذج المستعمل في معظم لغات البرمجة لمناولة الملفّات هو نموذج مناسب لإنشاء كائنات NT واستعمالها. إن الخصائص المتعلّقة بنموذج الملف هي:

- في معظم لغات البرمجة، قبل التمكُّن من قراءة ملفّ أو الكتابة إليه، يجب فتحه أولاً، ويمكن أن تكون عمليّة الفتح إما فتح ملفّ موجود أو إنشاء ملفّ جديد بإسم تحدّده. ويمكن أن يشمل إسم الملفّ دليلاً (أو تسلسل الأدلّة الهرمي) حيث يخزّن الملفّ.
- عند فتح ملف، حدّد العمليّات المطلوب تنفيذها ـ مثلاً، قراءة ملف أو الكتابة إليه أو الإلحاق إليه.
- نظام الملفّ يفتح الملفّ ويرجع مقبض ملفّ، يستعمل في عمليات لاحقة للملفّ المفتوح. وعند الإنتهاء من الملفّ، أغلق مقبض الملفّ.
- يشارك برنامجا ملفّ عندما يفتحان مقابض إليه في نفس الوقت. تتيح أيضاً بعض أنظمة الملفّات للتطبيقات إنشاء ملفّات مؤقّتة يحذفها نظام الملفّات تلقائياً عند إغلاق كل مقابضها.

وبواسطة بعض المناورات، يقلّد نموذج الكائن Windows NT نموذج الملفّ. أما الإختلافات الرئيسيّة فهي أن المقابض التي تسمى مقابض الكائن والكائنات المخزّنة في الذاكرة عِوَضاً عن الجهاز الفعلي. يوفّر القسم التالي بعض التفاصيل حول نموذج الكائن NT.

2-1-1-3 نموذج الكائن NT

مثل معظم أنظمة التشغيل، يستعمل النظام Windows NT المعالجات كجزء من العمل. ويخصّص لكل معالجة مجموعة من الموارد التي تتيح لها القيام بعمل معيّن:

شعبة تستطيع تنفيذ البرامج وفسحة عنوان لتخزين الشيفرة والبيانات، وعندما تشتغل الشعبة، فإنها تطلب موارد إضافيّة لمعالجتها عن طريق إنشاء كائنات أو عن طريق فتح مقابض إلى الكائنات الموجودة. إن مقابض الكائن فريدة المعالجة وتمثّل وصول المعالجة إلى موارد النظام. ويمكن إستعمالها لإستدعاء خدمات الكائن المحليّة التي تتناولها الموارد.

إن النظام الفرعي Win 32 هو معالجة NT، يعمل كملقّم لتطبيقات Win 32. وعندما يستدعي تطبيق روتين Win 32 API ينشىء كائناً بطريقة مباشرة أو غير مباشرة، يستدعي النظام الفرعي Win 32 خدمة كائن NT. ويأخذ برنامج إدارة الكائن NT زمام الحكم من هناك، حيث ينفّذ الوظائف التالية:

- تخصيص ذاكرة للكائن.
- إلحاق واصف أمان بالكائن، يحدّد المستعمل المتاح له إستعمال الكائن وما يمكن القيام به.
- إنشاء بنية دليل كائن وصيانتها حيث تخزّن أسماء الكائنات.
- إنشاء مقبض كائن وإرجاعه إلى المستدعي.

يجب أن تملك كل المعالجات في نمط المستعمل، بما فيها الأنظمة الفرعيّة للمحيط، مقبض إلى كائن قبل أن تتمكّن شعبها من إستعمال الكائن. غير أن إستعمال المقابض لمناولة موارد النظام ليست بفكرة جديدة. فمكتبات وقت التشغيل في اللغة C و Pascal (ولغات أخرى) على سبيل المثال، ترجع مقابض للملفّات المفتوحة. وبشكل مشابه، تستعمل تطبيقات Win 32 أنواع مختلفة من المقابض للتحكّم بالنوافذ ومؤشّر الماوس والرموز. وفي كلتا الحالتين، تستخدم المقابض كمؤشرات غير مباشرة إلى موارد النظام. يمنع عدم التوجُّه هذا البرامج التطبيقيّة من العبث مباشرة ببنيات بيانات النظام.

في البرنامج التنفيذي NT، توفّر مقابض الكائن فوائد إضافيّة. أولاً، وباستثناء ما تشير إليه، لا يوجد فرق بين مقبض ملفّ ومقبض الحدث ومقبض معالجة. ولا حاجة لتذكّر عشر آليات مختلفة لاستعمال عشرة أنواع مختلفة من الكائنات. ثانياً، يحتوي برنامج إدارة الكائنات على الحق الحصري في إنشاء المقابض وفي تحديد موقع كائن، يشير إليه الكائن. وهذا يعني أنه يمكن التدقيق في كل ما فعل في نمط المستعمل يؤثّر على كائن، بواسطة برنامج إدارة الكائنات. يتيح تأثير التبويب هذا تحقيق برنامج إدارة الكائنات لثلاثة أهداف تصميميّة رئيسيّة للنظام Windows NT:

- أنه يحمي الكائنات. ففي كل مرة تستعمل شعبة مقبض، ينفّذ برنامج إدارة الكائنات تدقيق أمان لمنح صلاحيّة للشعبة في إستعمال الكائن بالطريقة التي تحاول القيام بها.
- يراقب مستعمل الكائن. بحيث يستطيع حذف الكائنات المؤقتة عند عدم الحاجة لها. لكن برنامج إدارة الكائنات لا يحذف كائناً خلال إحتواء معالجة لمقبض إليه (أو خلال إحتواء النظام المؤشر إليه).
- يراقب إستعمال الموارد. وفي كل مرة تفتح شعبة مقبض كائن، يحدّد برنامج إدارة الكائنات لمعالجة الشعبة كلفة للذاكرة الفعليّة التي يستعملها الكائن. ولا يمكن أن يتجاوز إستعمال الموارد من قبل شعَب العمليّة حدود الذاكرة (الحصة) التي يحدّها مدير النظام للمستعمل الممثّل بالمعالجة.

إن حماية الكائنات هي جوهر المهمّة الأولى لنظام الأمان في Windows NT. وتستعيد إستعمالها بكثرة من نموذج الملفّ وهي مرئيّة للبرامج التطبيقيّة التي تستعملها روتينات Win 32 API. إن ما يلي هو مقدّمة موجزة لحماية الكائن ضمن البرنامج التنفيذي NT، وهو موضوع سيتمّ شرحه لاحقاً في هذا الفصل.

بالعودة إلى تمثيل الملفّ: عند فتح الملفّ، يجب تحديد ما سينفذ عليه لجهة القراءة أو الكتابة. فإذا حاولت الكتابة إلى ملفّ مفتوح للقراءة، ستحصل على رسالة خطأ. ويشكل مشابه في البرنامج التنفيذي NT، عندما تنشىء معالجة كائن أو تفتح مقبضاً إلى كائن موجود، يجب أن تحدّد المعالجة مجموعة من حقوق الوصول المطلوبة ـ أي، ما تريد تنفيذه على الكائن. ويمكن أن تطلب إما مجموعة من حقوق الوصول القياسيّة (مثل القراءة والكتابة والتنفيذ) المطبّقة على كل أنواع الكائنات أو تحدّد حقوق وصول تتغيّر وفقاً لنوع الكائن. فمثلاً، قد تطلب المعالجة الوصول لحذف كائن ملفّ أو الوصول لإلحاقه. ويشكل مشابه، قد تطلب القدرة على تعليق كائن شعبة أو إنهائه.

عندما تفتح معالجة مقبض إلى كائن، يستدعي برنامج إدارة الكائنات مراقب مراجع الأمان، وهو قسم نمط النواة لنظام الأمان، بحيث يرسل إليه مجموعة حقوق الوصول المطلوبة من قبل المعالجة. يدقّق مراقب مراجع الأمان لجهة سماح واصف الأمان العائد للكائن لنوع الوصول الذي تطلبه المعالجة. فإذا كان مسموحاً به، يرجع مراقب الأمان مجموعة حقوق الوصول الممنوحة المتاحة للمعالجة ويخزّنها برنامج إدارة الكائنات في مقبض الكائن الذي ينشئه.

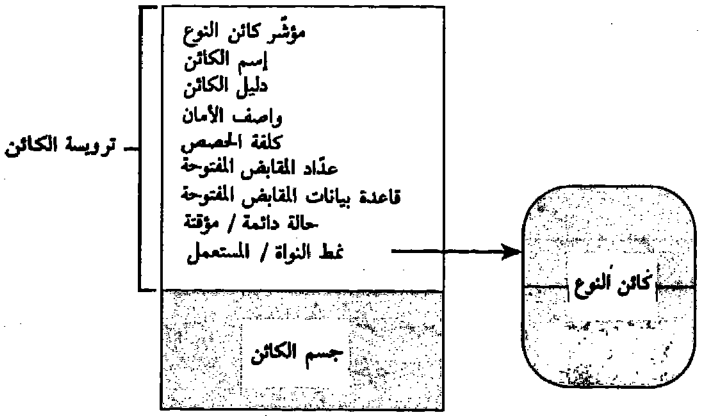
بعد ذلك، وعندما تستعمل شعَب المعالجة المقبض، يدقّق بسرعة برنامج إدارة الكائنات بتوافق مجموعة حقوق الوصول الممنوحة المخزّنة في المقبض مع الإستعمال المطبّق من قِبَل خدمة الكائن التي إستدعتها الشعَب. فمثلاً، إذا طلب المستدعي الوصول للقراءة إلى كائن مقطع لكنه إستدعى بعد ذلك الخدمة للكتابة إليه، فإن الخدمة تخفق. إن كيفيّة تحديد نظام الأمان لحقّ الوصول إلى الكائنات هو موضوع الشرح في القسم 3-3.

يتمّ شرح المهمّتين الثانية والثالثة التي تسهّلها مقابض الكائن ـ إحتجاز الكائن وإحتساب الموارد ـ في القسمين: 3-2-2-1 و 3-2-2-2.

3-1-2 بنية الكائن:

إن كل كائن NT هو نوع كائن معينّ. فالنوع يحدّد البيانات التي يحتويها الكائن وخدمات النظام المحليّة الممكن تطبيقها على الكائن. ولإدارة الكائنات المختلفة بشكل متناسق، يطلب برنامج إدارة الكائنات من كل كائن أن يحتوي عدّة مجالات من المعلومات القياسيّة في موقع

معروف. وباستمرار تواجد هذه البيانات، لا يعرف برنامج إدارة الكائنات ولا يهتمّ للبيانات الأخرى المخزّنة في الكائن. يحتوي كل كائن على قسمين ــ ترويسة الكائن وجسم الكائن ــ يفصلان البيانات القياسيّة للكائن عن البيانات المتغيّرة. ويتحكّم برنامج إدارة الكائنات بترويسة الكائن وتتحكّم المكوّنات التنفيذيّة الأخرى بأجسام الكائن من أنواع الكائن التي تنشئها.



الشكل (3-1)
محتويات ترويسة الكائن

الصفة	الغرض
ــ إسم الكائن	يجعل الكائن مرئياً إلى المعالجات الأخرى للمشاركة
ــ دليل الكائن	يوفّر بنية تسلسليّة حيث تخزّن أسماء الكائنات
ــ واصف الأمان	يحدّد مستعمل الكائن وما يمكن تنفيذه عليه
ــ كلفة الحصص	يسرد كلفة الموارد المحدّدة للمعالجة عندما تفتح مقبضاً إلى كائن
ــ عدّاد المقابض المفتوحة	يعدُّ عدد المرّات التي فُتح فيها مقبض إلى الكائن
ــ قاعدة بيانات المقابض المفتوحة	تسرد المعالجات التي فتحت مقابض إلى الكائن
ــ حالة دائمة / مؤقّتة	تشير إلى إمكانيّة حذف إسم الكائن وموقع تخزينه عند عدم إستعمال الكائن
ــ نمط النواة / المستعمل	يشير إلى توفّر الكائن في نمط المستعمل
ــ مؤشّر كائن النوع	يشير إلى كائن النوع الذي يحتوي صفات عامة لمجموعة من الكائنات المشابهة

الجدول (3-2)
صفات ترويسة الكائن القياسيّة

يستعمل برنامج إدارة الكائنات البيانات المخزّنة في ترويسة كائن لإدارة الكائنات دون إعتبار لنوعها. يظهر الشكل (1-3) البيانات أو الصفات التي تحتويها كل ترويسات الكائن. ويصف الجدول (2-3) بإختصار صفات ترويسة الكائن.

يوفّر برنامج إدارة الكائنات مجموعة صغيرة من الخدمات العامة تعمل على الصفات المخزّنة في ترويسة كائن ويمكن إستعمالها على كائنات من أي نوع (رغم أن بعض الخدمات العامة لا تعني شيئاً لبعض الكائنات). تسرد في الجدول (3-3) هذه الخدمات العامة، والتي يوفّر النظام الفرعي Win 32 بعض منها إلى تطبيقات Win 32.

الخدمة	الغرض
ـ إغلاق	تغلق مقبضاً إلى كائن
ـ إستنساخ	تشارك كائناً عن طريق إستنساخ مقبضاً وتزويده إلى معالجة أخرى
ـ الإستعلام عن كائن	تجلب معلومات الصفات القياسيّة لكائن
ـ الإستعلام عن الأمان	تجلب واصف الأمان الكائن
ـ ضبط الأمان	تغيّر الحماية على كائن
ـ إنتظار كائن واحد	تزامن تنفيذ شعبة مع كائن واحد
ـ إنتظار عدّة كائنات	تزامن تنفيذ شعبة مع عدّة كائنات

الجدول (3-3)
خدمات الكائن العامة

إضافةً إلى ترويسة الكائن، يحتوي كل كائن على جسم كائن ذات نسق ومحتويات خاصة بنوع الكائن، وكل الكائنات من نفس النوع تشارك نفس نسق جسم الكائن. وعن طريق إنشاء نوع كائن وتزويده بالخدمات، يستطيع مكوّن تنفيذي التحكّم بمناولة البيانات في كل أجسام الكائنات من ذلك النوع.

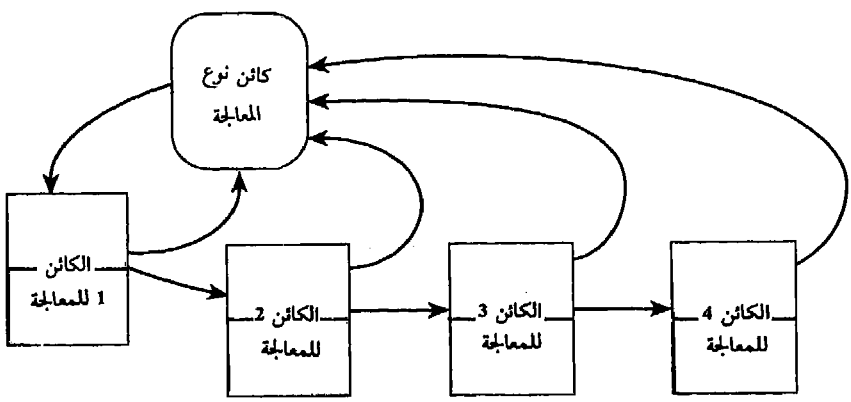
يستطيع كل مكوّن برنامج تنفيذي NT تعريف أنواع الكائنات. يتألف تعريف نوع كائن من تحديد البيانات التي تُخزّن في جسم كل حالة آنيّة للنوع الجديد، حيث يبلغ برنامج إدارة الكائنات عن حجم الجسم لكي يتمكّن من تخصيص كميّة ذاكرة مناسبة عندما يتمّ إنشاء الكائنات وتزويد الخدمات لنوع الكائن الجديد. فمثلاً، يعرّف برنامج إدارة الكائنات جسم

كائنات المعالجة ويوفّر الخدمات المحليّة التي تتناول البيانات المخزّنة هناك. وبشكل مشابه، يعرّف برنامج إدارة الدخل / الخرج محتويات جسم كائن ملفّ ويصدّر الخدمات التي تجلب هذه البيانات أو تضبطها. يتمّ وصف محتويات أجسام الكائنات المختلفة لاحقاً في هذا الكتاب سويّة مع مكوّنات البرنامج التنفيذي NT التي تعرّفها.

3-1-3 أنواع الكائنات:

تحتوي ترويسات الكائنات على بيانات التي هي عامة لكل الكائنات والتي يمكن أن تكون بقيم مختلفة لكل حالة آنيّة لكائن. فمثلاً، يحتوي كل كائن على إسم خاص به ويمكن أن يحتوي على واصف أمان خاص به. لكن الكائنات يمكن أن تحتوي بعض البيانات التي تبقى ثابتة لكل الكائنات من نوع معيّن. فمثلاً، يمكن الإنتقاء من مجموعة حقوق الوصول الخاصة بنوع كائن عند فتح مقبض إلى كائنات من نوع معيّن. ويزوّد البرنامج التنفيذي NT الوصول للإنهاء والتعليق لكائنات الشعبة والوصول لقراءة كائنات الملفّات وكتابتها وإلحاقها وحذفها. مثال آخر لصفة خاصة بنوع كائن هي المزامنة، أي قدرة شعبة على إنتظار ضبط كائنات من نوع معيّن إلى الحالة المُشار إليها، والتي سيتمّ شرحها لاحقاً.

لحفظ الذاكرة وتخفيض الصيانة، يضبط برنامج إدارة الكائنات الصفات الستاتيّة الخاصة بنوع كائن عند إنشاء نوع كائن جديد. وهو يستعمل كائن من نوعه يسمى كائن النوع لتسجيل هذه البيانات. وكما يوضّح الشكل (3-2) على الصفحة التالية، يربط كائن نوع أيضاً كل الكائنات من نفس النوع سويّة حيث يتيح لبرنامج إدارة الكائنات إيجادها وعدّها عند الضرورة.



الشكل (3-2)
كائنات المعالجة وكائن نوع المعالجة

يمكن مناولة كائنات النوع من نمط المستعمل لأن برنامج إدارة الكائنات لا يزوّد خدمات إليها. لكن بعض الصفات التي تعرّفها مرئيّة عبر بعض الخدمات المحليّة المحدّدة وعبر روتينات Win 32 API. يتمّ وصف الصفات المخزّنة في أجسام كائنات النوع في الجدول (3-4).

الصفة	الغرض
ـ إسم نوع الكائن	إسم الكائنات من هذا النوع («المعالجة» و«الحدث» و«المنفّذ» وما شابه).
ـ أنواع الوصول	أنواع الوصول التي تستطيع الشعبة طلبها عند فتح مقبض إلى كائن من هذا النوع («القراءة» و«الكتابة» و«الإنهاء» و«التعليق» وما شابه).
ـ قدرة المزامنة	إمكانيّة إنتظار شعبة للكائنات من هذا النوع.
ـ قابلة التعيين صفحات / غير قابلة لتعيين صفحات	إمكانيّة الكائنات من هذا النوع تعيين صفحات في الذاكرة
ـ الطرق	روتين واحد أو أكثر يستدعيه برنامج إدارة الكائنات تلقائياً عند نقاط معيّنة خلال مدة خدمة الكائن

الجدول (3-4)
صفات نوع الكائن

يعود التزامن وهو أحد الصفات المرئيّة لتطبيقات Win 32، إلى قدرة شعبة على مزامنة تنفيذها بواسطة إنتظار تغيّر كائن من حالة إلى أخرى. تستطيع الشعبة أن تتزامن مع كائنات المعالجة التنفيذية والشعبة والملفّ وزوج أحداث والإعلام الإشاري والخوافت الموقّت. ولا تدعم المزامنة كائنات المقطع والمنفّذ وصفة الوصول ودليل الكائن والربط الرمزي والاستمثال والمرجعيّة.

تتألف الصفة الأخيرة في القائمة ـ الطرق ـ مجموعة من الروتينات الداخليّة المشابهة للوظائف الإنشائيّة والإتلافيّة في اللغة C++ أي الروتينات التي يتمّ إستدعاءها تلقائياً عند إنشاء كائن أو إتلافه. يوسّع برنامج إدارة الكائنات هذا المبدأ بواسطة إستدعاء طريقة كائن في حالات أخرى أيضاً، مثل عند فتح مقبض إلى كائن أو إغلاقه أو عند محاولة تغيير الحماية على كائن. تحدّد بعض أنواع الكائنات الطرق بينما لا يقفل ذلك البعض الآخر وفقاً لكيفيّة إستعمال الكائن. يتمّ وصف الطرق (والتي تسمى أحياناً الطرق الظاهريّة) في القسم 3-2-3.

بإيجاز، تتألف كائنات البرنامج التنفيذي NT من قسمين: ترويسة كائن المحكومة من قِبَل برنامج إدارة الكائنات وجسم الكائن المحكوم بواسطة مكوّنات نظام التشغيل التي تنشىء نوع كائن. إحدى الصفات في ترويسة الكائن هي مؤشّر كائن النوع وهي مبنيّة تعرّف الصفات الستاتيّة للكائنات من النوع الجديد. ويستطيع أي مكوّن في البرنامج التنفيذي NT تعريف أنواع الكائنات الجديدة ويزوّد كل مكوّن الخدمات لأنواع الكائن التي تعرّفها.

2-3 الكائنات:

كما سبق وذُكر، يوفّر برنامج إدارة الكائنات مجموعة من الخدمات العامّة التي تعمل على كل أنواع الكائنات. إضافة لذلك، تزوّد المكوّنات الأخرى للبرنامج التنفيذي NT خدمات خاصة لنوع الكائن لأنواع الكائنات التي تنشئها. وهكذا، يجب أن تمر كل الخدمات التي تتناول كائناً عبر برنامج إدارة الكائنات على مستوى واحد أو آخر. وهذا يتيح لبرنامج إدارة الكائنات مركّزة الحكم على الكائنات وتنفيذ كل مهام إدارة الكائنات (أو التخلّي على التحكُّم لصالح برنامج إدارة كائنات آخر).

يركّز هذا القسم على الوظائف الأوليّة لبرنامج إدارة الكائنات. يطبّق القسمان الفرعيّان الأوّلان كيفيّة تحديده لموقع الكائنات وكيفيّة إرساله المقابض إليها. ويشرح القسم الفرعي الثالث بتفصيل طرق الكائنات. وهذه الكائنات والخدمات الموصوفة مرئيّة للأنظمة الفرعيّة في نمط المستعمل ما لم يذكر خلاف ذلك.

1-2-3 أسماء الكائنات:

إن أهم إعتبار في إنشاء عدّة كائنات هو توريث نظام ناجح لمتابعة تعقُّبها. يتطلّب برنامج إدارة الكائنات ما يلي للقيام بذلك:

- طريقة لتفريق كائن عن آخر.
- طريقة لإيجاد كائن معيّن وإسترداده.

يلبّى المطلب الأول بواسطة الإتاحة بتعيين الأسماء إلى الكائنات. وهذا إمتداد لما توفّره معظم أنظمة التشغيل ــ القدرة على تسمية موارد منتقاة أو ملفّات أو أنابيب أو كتلة من الذاكرة المشاركة. بينما من الناحية المقابلة، يتيح البرنامج التنفيذي NT لأي مورد ممثّل بكائن أن يحتوي على إسم.

كذلك، يلبّى المطلب الثاني، إيجاد كائن، بواسطة أسماء الكائنات. فإذا قام برنامج إدارة الكائنات بتخزين الكائنات وفقاً لأسمائها، فإنه يستطيع إيجاد كائن بواسطة البحث عن إسمه.

كذلك تلبّي أسماء الكائنات المطلب الثالث، حيث تتيح للمعالجات مشاركة الكائنات. إن فسحة إسم الكائن للبرنامج التنفيذي هي فسحة عامة ومرئيّة لكل المعالجات في النظام. تستطيع معالجة واحدة إنشاء كائن ووضع إسمه في فسحة الإسم العامة، وتستطيع معالجة ثانية فتح مقبض إلى الكائن عن طريق تحديد إسم الكائن. وإذا لم يرد أن يشارك الكائن بهذه الطريقة، فلا حاجة لإعطائه إسماً من قِبَل الناشىء.

لزيادة الكفاية، لا يبحث برنامج إدارة الكائنات عن إسم الكائن في كل مرة يستعمل الكائن، لكنه يبحث عن إسم بظلّ حالتين فقط. الأولى، عندما تنشىء معالجة كائن بإسم: يبحث برنامج إدارة الكائنات عن الإسم ليتأكد من عدم وجوده سابقاً قبل تخزين الإسم الجديد في فسحة الإسم العامة. الثانية، عندما تفتح معالجة مقبض إلى كائن بإسم؛ يبحث برنامج إدارة الكائنات عن الإسم ويجد الكائن ثم يرجع مقبض كائن إلى المستدعي، حيث يستعمل المستدعي المقبض للعودة إلى الكائن. وعند البحث عن الإسم، يتيح برنامج إدارة الكائنات للمستدعي إنتقاء بحث حسّاس لحالة الأحرف أو بحثاً غير حسّاس لحالة الأحرف وهي مزيّة يدعمها النظام POSIX والمحيطات الأخرى التي تستعمل أسماء ملفّات حسّاسة كآلة الأحرف.

إن أسماء الكائنات عامة لحاسوب واحد (أو لكل المعالجات على حاسوب متعدّد المعالجات) لكنها غير مرئيّة عبر شبكة، لكن برنامج إدارة الكائنات يوفّر عقيفة ــ تسمّى طريقة التحليل اللغوي ــ للوصول إلى الكائنات بالأسماء الموجودة على حواسيب أخرى. فمثلاً، يمدّد برنامج إدارة الدخل / الخرج الذي يزوّد خدمات كائن الملفّ، وظائف برنامج إدارة الكائنات إلى الملفات البعيدة. وعندما يطلب منه فتح كائن ملفّ بعيد، يستدعي برنامج إدارة الكائنات طريقة التحليل اللغوي، والتي تتيح لبرنامج إدارة الدخل / الخرج إعتراض الطلب وتسليمه إلى موجّه الشبكة الجديد، وهو مسيق يوصّل إلى الملفّات عبر الشبكة. وتستدعي معالجة ملقّم على نظام Windows NT البعيد برنامج إدارة الكائنات وبرنامج إدارة الدخل / الخرج على ذلك النظام لإيجاد كائن الملفّ وإرجاع المعلومات عبر الشبكة. وقد تستعمل تمديدات النظام المستقبليّة نفس عقيفة برنامج إدارة الكائنات لإدارة الكائنات البعيدة الأخرى. (يتم وصف الطرق بتفصيل أكبر في القسم 3-2-3. ويتمّ وصف شبكات Windows NT في الفصل التاسع).

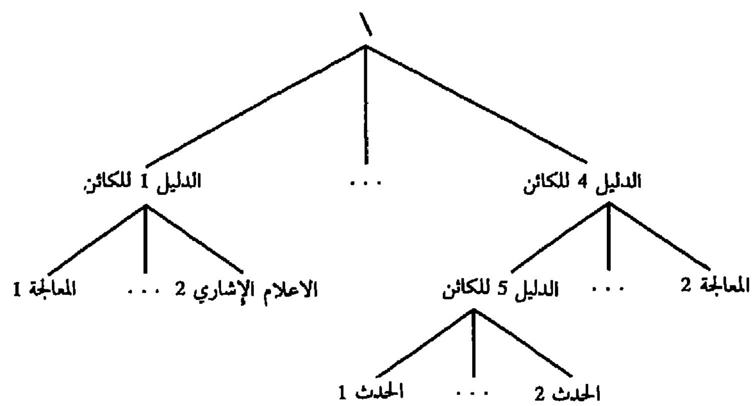
1-1-2-3 أدلّة الكائن:

خلال تقرير كيفيّة تشكيل أسماء الكائنات، واجهَ المطوِّرون تعقيدات أوليّة من قِبَل أنظمة الملفّات MS-DOS وPOSIX التي تحتوي على مخطّطات تسمية تسلسليّة هرميّة للملفّات وأدلّة الملفّات. وفي البرنامج التنفيذي NT، تمثّل الملفّات والأدلّة ككائنات ولذلك، يجب أن يستوعب برنامج إدارة الكائنات نسق أسماء الملفّات لإيجاد كائنات الملفّات. وهذا ما يجعل من أسماء الكائنات أسماء ملفّات تمثيليّة.

تحتوي أسماء كائنات NT على بعض خصائص أسماء الملفّات في MS-DOS وPOSIX. يصوّر الشكل 3-3 التسلسل الهرمي لإسم الكائن NT.

لاحِظْ أن جذر شجرة إسم الكائن هي الشرطة الخلفية (\) في MS-DOS. تمثّل عُقَد الأوراق على الشجرة الكائنات الإفراديّة وتمثّل العُقَد المتوسّطة أسماء أدلّة الكائنات. يتمُّ نسق أسماء الكائنات بدءاً من الجذر واستعراض المسارّ إلى كائن. وكما في MS-DOS وOS/2، تستعمل الشرطات الخلفية لفصل الأسماء في المسار.

إن كائن دليل الكائن هو وسائل برنامج إدارة الكائنات لدعم بنية التسمية التسلسليّة هذه، وهو متناظر مع دليل نظام الملفّات ويحتوي أسماء الكائنات الأخـرى، ويمكن أيضاً أن يحتوي أدلّة الكائنات الأخرى. يستطيع النظام الفرعي Win 32 والأنظمة الفرعيّة الأخرى وكذلك مكوّنات البرنامج التنفيذي NT، إنشاء تسلسلات هرميّة تحكُّميّة لأدلّة الكائنات حيث يتمُّ تخزين الكائنات المسمّاة التي تنشئها.



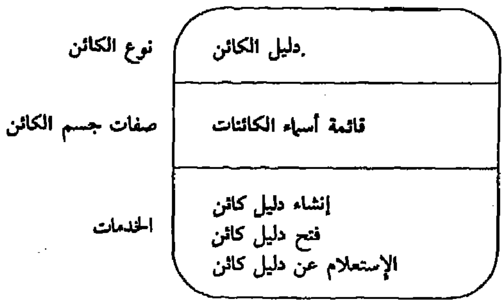
الشكل (3-3)
التسلسلي الهرمي لإسم الكائن

يُظهِرُ الشكل (4-3) على الصفحة التالية الملخّص المفهومي للخصائص المهمّة الفريدة لكائن دليل الكائن. وفي هذه المخططات وفي الأخرى الموجودة في هذا الكتاب، ينسب نوع الكائن إلى فئة الكائنات الموصوفة. وتنسب صفات جسم الكائن إلى مجالات البيانات المخزّنة في أجسام الكائنات من ذلك النوع. والخدمات هي خدمات النظام المحليّة التي توفّر مكوّنات البرنامج التنفيذي NT لمناولة صفات الكائن، (ولا تظهر الصفات المخزّنة في ترويسات الكائن لأنها هي نفسها لكل الكائنات من كل الأنواع. وبشكل مشابه، تتناول خدمات الكائن العامة الموصوفة سابقاً كائنات من كل الأنواع).

تستعمل خدمات الإنشاء والفتح لإنشاء أدلّة الكائن ولفتح مقابض إليها. وبعد أن تُفتح شعبة مقبض (بواسطة الوصول للكتابة) إلى دليل كائن، فإنها تنشىء كائنات أخرى وتصفها في دليل الكائن.

وتتيح خدمة الإستعلام لمستدعي مسح قائمة أسماء الكائنات المخزّنة في دليل الكائن. ويحتفظ كائن دليل الكائن بما يكفي من المعلومات لترجمة أسماء الكائنات هذه إلى مؤشّرات إلى الكائنات نفسها. يستعمل برنامج إدارة الكائنات المؤشّرات لإنشاء مقابض الكائنات والتي ترجعها إلى المستدعي في نمط المستعمل.

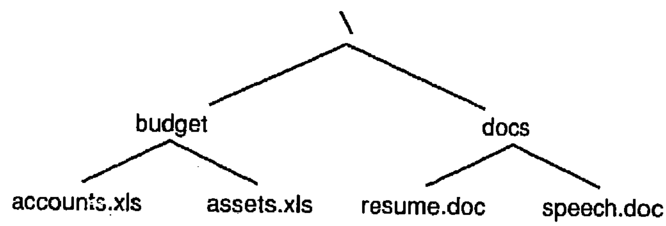
ويستطيع كل من نمط النواة ونمط المستعمل (مثل الأنظمة الفرعيّة) إنشاء أدلّة كائنات حيث تخزّن الكائنات. فمثلاً، ينشيء برنامج إدارة الدخل / الخرج دليل كائن يسمى \Device والذي يحتوي أسماء الكائنات التي تمثّل أجهزة الدخل / الخرج.



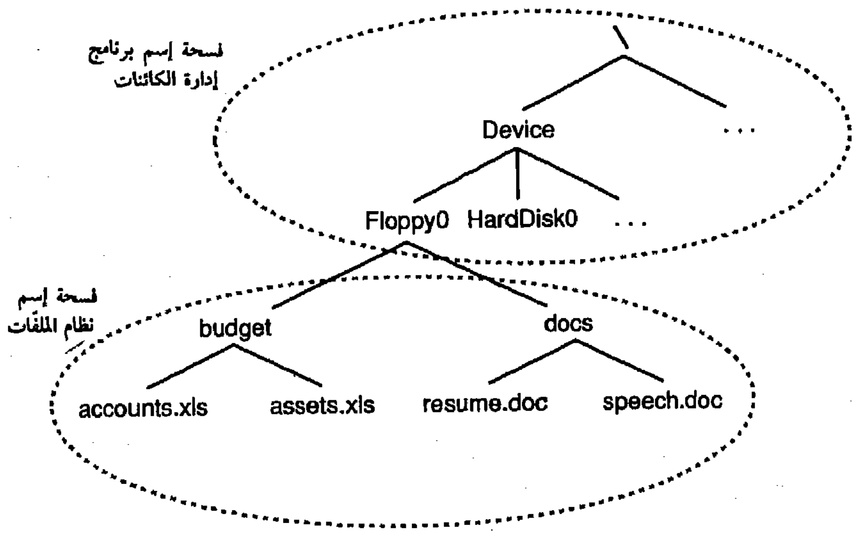
الشكل (4-3)
كائن دليل الكائن

تتكرّر خدمات الكائن الثلاث، الإنشاء والفتح والإستعلام في البرنامج التنفيذي NT. فبرنامج إدارة الدخل / الخرج يستخدم خدمة إنشاء ملفّ لكائنات ملفّاته ويستخدم برنامج

إدارة المعالجة خدمة إنشاء معالجة لكائنات معالجاتها. ورغم أن مطوّري NT إعتبروا إنشاء روتين البارامترات المطلوبة لتحفيز كائن ملفّ، مثلاً، تختلف عن البارامترات المطلوبة لتحفيز كائن معالجة. وقد يصبح روتيناً واحداً باللغة C أكثر تعقيداً إذا أُضيفت أنواع كائنات جديدة إلى النظام. كذلك، قد يجلب لنفسه برنامج إدارة الكائنات معالجات إضافيّة في كل مرّة تستدعي شعبة خدمة كائن لتحديد نوع الكائن الذي يشير إليه المقبض وعند إستدعاء الإصدار المناسب للخدمة. لهذه الأسباب ولأسباب أخرى، تستخدم خدمات الإنشاء والفتح والإستعلام بشكل مستقلّ لكل نوع كائن.



ضمن فسحة إسم برنامج إدارة الكائنات، تتّخذ بنية الدليل الشكل التالي:



في هذه الشجرة، يمثّل كل إسم كائن برنامج تنفيذي NT. وقد شملت فسحة إسم نظام الملفّ في فسحة إسم الكائن تحت الإسم \Device\Floppy 0.

2-1-2-3 حقول الكائن:

توفّر فسحة إسم الكائن مظلّة ينطوي تحتها مجموعات الكائنات الذاتيّة الاحتواء والتي تسمى حقول الكائنات، حيث تتيح بتمديد فسحة إسم الكائن. فمثلاً، فإن برنامج إدارة الدخل / الخرج هو برنامج إدارة كائنات ثانوي يتحكّم بجعل كائن ويتألف من ملفّات الأقراص وأدلّتها وأجهزتها. يتيح برنامج إدارة الكائنات لنظام الدخل / الخرج إدخال كائنات نظام الملفّات تحت عقدة ورق فسحة إسم برنامج إدارة الكائنات. إفترض، على سبيل المثال، أنك تحتوي على بُنية الدليل التالي على قرص مرن:

عندما يحاول مستعمل برنامج Excel for Windows من Microsoft فتح الملفّ A:\budget\accounts.xls، يفتح برنامج إدارة الكائنات مقبضاً إلى كائن الملفّ المسمّى \Device\Floppy0\budget\accounts.xls. للقيام بذلك، يبحث برنامج إدارة الكائنات عن فسحة إسمه إلى أن يبلغ الكائن المسمّى Floppy0 الذي هو كائن جهاز خاص يحتوي على طريقة تحليل لغوي خاصة به. يعلّق برنامج إدارة الكائنات بحثه عن الإسم ويستدعي طريقة التحليل اللغوي، حيث يمرّر إليها الإسم \budget\accounts.xls. تزوّد طريقة التحليل اللغوي من قِبَل نظام الدخل / الخرج الذي يطلب من نظام الملفّات الصحيح تحديد موقع هذا الملفّ المخزّن على القرص المرن وفتحه. يتمّ وصف الطرُق بتفضيل أكبر في القسم 3-2-3.

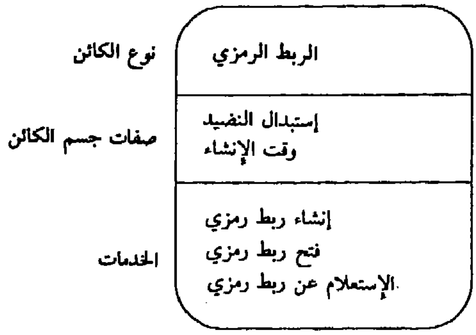
3-1-2-3 الروابط الرمزيّة:

في أنظمة الملفّات المعيّنة (على بعض أنظمة UNIX مثلاً) يتيح الربط الرمزي للمستعمل إنشاء إسم ملفّ أو إسم دليل والذي يترجم، عند إستعماله، من قبل نظام التشغيل إلى إسم دليل أو ملفّ مختلف. وهي طريقة بسيطة تتيح للمستعمل المشاركة غير المباشرة في ملفّ أو في محتويات دليل وإنشاء ربط متقاطع بين الأدلّة المختلفة في بنية الدليل التسلسليّة الهرميّة العاديّة.

يستخدم برنامج إدارة الكائنات NT كائناً يسمّى كائن ربط رمزي، ينفّذ وظيفة مشابهة لأسماء الكائنات في فسحة إسم كائنها. وعندما يشير مستدعي إلى إسم كائن ربط رمزي، يستعرض برنامج إدارة الكائنات فسحة إسم الكائن إلى أن يبلغ كائن الربط الرمزي. وهو يبحث داخل الربط الرمزي ويجد نضيداً يستبدله في إسم الربط الرمزي. ثم يعاود البحث عن إسمه. يمكن أن يحصل الربط الرمزي في أي مكان ضمن نضيد إسم كائن. يلخّص الشكل (3-5) صفات وخدمات نوع كائن الربط الرمزي.

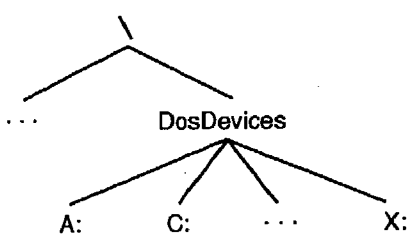
إحدى الأماكن حيث يستعمل البرنامج التنفيذي NT كائنات الربط الرمزي هي في ترجمة

أسماء أجهزة MS-DOS إلى أسماء كائنات Windows NT. وفي MS-DOS، يشير المستعمل إلى سوّاقات القرص المرن والقرص الصلب بإستعمال الأسماء :A و :B و :C وما شابه. إضافة لذلك، يستطيع المستعمل إضافة أسماء سوّاقة جديدة أو سوّاقة زائفة عن طريق إنشاء تقسيمات إضافيّة على قرص صلب أو بتعريف إسم سوّاقة ليشير إلى دليل قرص على حاسبة آخر. وبعد إنشائها، يجب أن تكون أسماء السوّاقات هذه مرئيّة لكل المعالجات على النظام.

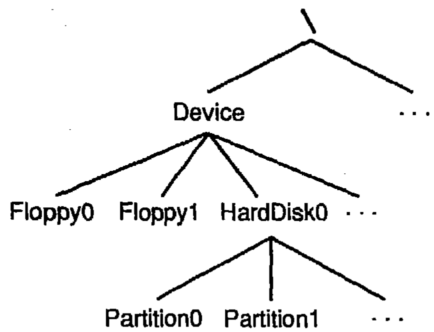


الشكل (3-5)
كائن الربط الرمزي

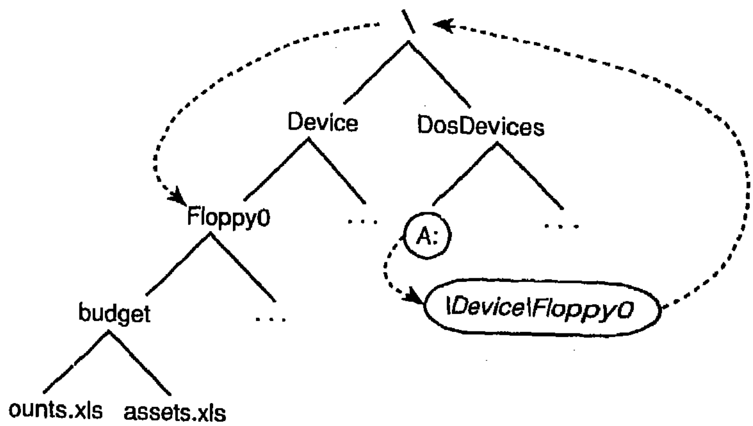
يحمي النظام الرمزي Win 32 أحرف السواقة، وهي بيانات عامة عن طريق وضعها في فسحة إسم برنامج إدارة الكائنات. ويتمّ إنشاء دليل كائن خاص لهذا الغرض، كما يظهر هنا:



وعندما ينشىء مستعمل أو برنامج تطبيقي حرف سواقة جديد، يضيف النظام الفرعي Win 32 كائن آخر تحت دليل الكائن DosDevices\. لكن الكائنات التي تمثّل الأجهزة الفعليّة تخرج من مكان آخر في الشجرة، كما يرسم هنا:



إن الكائنات المسمّاة :A و:B و:C وما شابه هي كائنات ربط رمزيّة ويحتوي كل من هذه الروابط الرمزيّة على إسم كائن الجهاز الفعلي حيث يشير حرف السوّاقة. لذلك، وبالنسبة للمثال، إذا فتح مستعمل البرنامج Excel for Windows الصفحة الجدوليّة المخزّنة في A:\budget\accounts.xls . يترجم النظام الغربي Win 32 هذا الاسم ويفتح مقبض إلى كائن الملف المسمى DosDevice\A:\budget\accounts.xls\. ولإيجاد كائن الملفّ هذا، يستعرض برنامج إدارة الكائنات شجرة إسم الكائن إلى أن يبلغ الكائن المعروف بإسم :A ويكتشف أن هذا الكائن هو ربط رمزي. وهو يدقّق بمحتويات كائن الربط الرمزي حيث يجد النضيد Device\Floppy0\ المخزّن في الداخل، كما يظهر هنا:



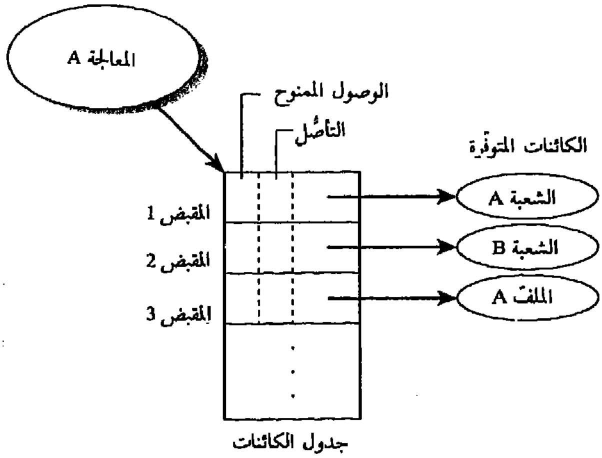
يأخذ برنامج إدارة الكائنات النضيد المخزّن في كائن الربط الرمزي ويلحق بقيّة النضيد الأصلي معه (Device\Floppy0 plus\budget\accounts.xls\). ثم يعاود البحث عن كائن الملفّ من أعلى الشجرة.

تتيح الروابط الرمزيّة لنظام فرعي (أو شيفرة أخرى) إنشاء أسماء مستعارة للكائنات التنفيذيّة والتي يستطيع النظام الفرعي تغييرها عندما يريد. إضافة لذلك، يستطيع النظام الفرعي التعرُّف إلى كسب الأداء بتخزين البيانات العامّة مثل أسماء السوّاقات، مباشرة في البرنامج التنفيذي بدلاً من فسحة عنوان النظام الفرعي. يشرح موضوع أداء النظام الفرعي بالتفصيل في الفصل الخامس «Windows والأنظمة الفرعيّة المحميّة».

2-2-3 مقابض الكائن:

رغم أن أسماء الكائن مهمّة لتخزين الكائنات ولمشاركة الكائنات، لكنها لا تستعمل في غالب الأحيان. تحدّد معالجة إسم كائن عندما تنشىء أولاً الكائن أو عندما تفتح مقبضاً إليه. بعد ذلك، تستعمل المعالجة مقبض الكائن. إن العودة إلى كائن بواسطة مقبضه أسرع من إستعمال إسمه لأن برنامج إدارة الكائنات يستطيع تجاوز البحث عن الإسم وإيجاد الكائن مباشرة.

إن مقبض كائن NT هو فهرس في جدول كائنات خاص للمعالجة. يحتوي جدول كائن المعالجة مؤشرات إلى كل الكائنات التي فتحت إليها المعالجة مقبض. وتحدّد المعالجة مقابض لكائنات عن طريق إنشاء كائن وبواسطة فتح مقبض إلى كائن موجود، وبواسطة تأصُّل مقبض من معالجة أخرى، أو بإستلام مقبض مستنسخ من معالجة أخرى. يوضّح الشكل (3-6) العلاقة بين معالجة وجدول الكائنات العائد لها.



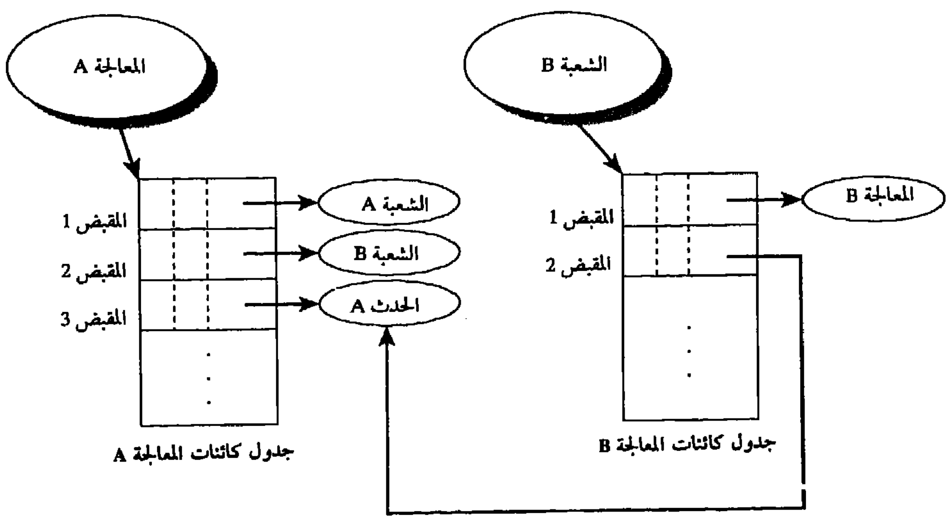
الشكل (3-6)
بنية جدول الكائنات

يحتوي كل إدخال في جدول الكائنات على حقوق وصول ممنوحة للمقبض الموافق وتسميته المتأصّلة ـ أي، حصول المعالجات التي أُنشئت بواسطة هذه المعالجة على نسخة المقبض في جداول كائناتها. ورغم أن التسمية مقبض يعود إلى الفهرس في الجدول، يستعمل المطوّرون التسمية مقبض ليشيروا أيضاً إلى البيانات المخزّنة في إدخال الجدول الموافق.

تشارك معالجتان كائن عندما يفتحان مقابض إليه. ويكون المقبضان فريدان، كما يوضح ذلك الشكل (3-7) على الصفحة التالية.

يحدّد منشىء الكائن إمكانيّة تأصُّل المقابض إلى الكائن من معالجة واحدة بواسطة المعالجات التي تنشئها. تدعم هذه المزيّة هذه المحيطات بما فيها Win 32 و POSIX، التي تتيح تأصُّل الموارد.

وعند إنهاء معالجة، يصبح كائن المعالجة عرضة للحذف من النظام (وفقاً لاستعماله من قبل أية معالجة أخرى، كما سيشرح لاحقاً). وقبل حذف كائن معالجة، يستدعي برنامج إدارة الكائنات طريقة الحذف لكائنات المعالجة التي تغلق كل المقابض في جدول كائنات المعالجة. (راجع القسم 3-2-3 لمزيد من المعلومات).



الشكل (3-7)
مشاركة كائن

1-2-2-3 إحتجاز الكائن:

لأنه يجب على كل المعالجات في نمط المستعمل التي توصل إلى كائن، أن تفتح أولاً مقبض إليه، يستطيع برنامج إدارة الكائنات تعقُّب عدد هذه المعالجات وحتى تلك التي تستعمل الكائن. يمثّل تعقُّب هذه المقابض الخطوة الأولى في تطبيق إحتجاز الكائن ـ أي، إحتجاز الكائنات المؤقتة فقط طالما أنها تستعمل ثم تحذف.

يستخدم برنامج إدارة الكائنات إحتجاز الكائن في مرحلتين. المرحلة الأولى تسمى إحتجاز الإسم، وهي محكومة بواسطة عدد المقابض المفتوحة لكائن. وفي كل مرة تفتح معالجة مقبض إلى كائن، يزيد برنامج إدارة الكائنات عدّاد المقابض المفتوحة في ترويسة الكائن. (راجع الشكل (3-1) على الصفحة 62). وعند المعالجات من إستعمال الكائن وإغلاق مقابضها إليه، يخفّض برنامج إدارة الكائنات العدّاد. وعندما يصبح العدّاد صفراً، يحذف برنامج إدارة الكائنات إسم الكائن من فسحة الإسم العامة. وهذا يمنع المعالجات الجديدة من فتح مقبض إلى الكائن. (لا تحذف أسماء الكائنات الدائمة لأن هذه الكائنات تمثّل الكينونات مثل الأجهزة الفعليّة، التي تبقى متواجدة حتى عند عدم إستعمالها من قِبَل المعالجة. ويجب أن يغيّر نظام التشغيل الكائنات الدائمة إلى كائنات مؤقتة قبل أن يتمكّن من حذفها).

المرحلة الثانية لاحتجاز الكائن هي في إيقاف كائنات الإحتجاز (أي لحذفها) عند عدم إستعمالها. ولأن نظام التشغيل يوصل عادة إلى الكائنات باستعمال المؤشرات بدلاً من المقابض، يجب على برنامج إدارة الكائنات أن يسجّل أيضاً عدد مؤشرات الكائنات التي وزّعتها على معالجات نظام التشغيل. وهو يزيد عد مرجعي لكائن في كل مرة يعد فيها مؤثراً إلى الكائن. وعندما تنتهي شعَب نظام التشغيل من إستعمال المؤشّر، فإنها تستدعي برنامج إدارة الكائنات لتخفيض العدُّ المرجعي للكائن. وحتى بعد أن يبلغ عدّاد المقابض المفتوح لكائن الصفر، قد يبقى العدُّ المرجعي للكائن مرجعيًّا، حيث يشير إلى أن نظام التشغيل ما زال يستعمل الكائن. وأخيراً يهبط العدُّ المرجعي أيضاً إلى الصفر. وعندما يحصل ذلك، يحذف برنامج إدارة الكائنات الكائن في الذاكرة.

وبسبب طريقة عمل إحتجاز الكائنات، يستطيع برنامج الكائنات ضمان بقاء الكائن وإسمه في الذاكرة عن طريق إبقاء المقبض مفتوحاً إلى الكائن. ولا حاجة لأن يهتمّ المبرمجون الذين يكتبون برامج تطبيقيّة تحتوي معالجتين أو أكثر متوافقة لجهة حذف معالجة واحدة لكائن قبل إنتهاء المعالجة التالية من إستعماله. إضافة لذلك، لن تسبّب مقابض كائن البرنامج التطبيقي في حذف كائن إذا كان نظام التشغيل يستعمله. فمثلاً قد تنشىء معالجة واحدة معالجة ثانية لتنفيذ برنامج في الخلفيّة. ثم تغلق فوراً مقبضها إلى المعالجة. ولأن نظام التشغيل يحتاج

لمعالجة ثانية لتشغيل البرنامج، فإنه يحافظ على مرجع لكائن المعالجة. وفقط وعند إنتهاء تنفيذ برنامج الخلفيّة يقدّم برنامج إدارة الكائنات بتخفيض العدُّ المرجعي للمعالجة الثانية ثم حذفها.

2-2-2-3 حساب الموارد:

يتعلّق حساب الموارد، مثل إحتجاز الكائنات، باستعمال مقابض الكائنات. فإذا كان الكائن يتّصف بعد مقابض مفتوحة موجب، فإنه يشير إلى إستعمال بعض المعالجات لهذه الموارد. ويشير أيضاً إلى أنه حدّد كلفة لهذه المعالجة للذاكرة التي يشغلها الكائن. وعندما يهبط عدُّ مقابض الكائن إلى الصفر، لا تحدّد كلفة للمعالجة التي كانت تستعمل الكائن.

يستعمل العديد من أنظمة التشغيل نظام الحصص لتحديد وصول المعالجات إلى موارد النظام. لكن أنواع الحصص المفروضة على المعالجات هي في بعض الأحيان مختلفة ومعقّدة وتنشر شيفرة تعقُّب الحصص في كل نظام التشغيل. فمثلاً، وفي بعض أنظمة التشغيل، قد يسجّل مكوّن دخل / خرج ويحدّد عدد الملفّات التي تستطيع المعالجة فتحها، بينما يفرض مكوّن ذاكرة حدّاً على كميّة الذاكرة التي تخصّصها شعَب المعالجة. وقد يحدّ مكوّن معالجة المستعمل إلى عدد أقصى من المعالجات الجديدة الممكن إنشاؤها أو إلى عدد أقصى من الشعَب ضمن المعالجة. ويتمُّ تعقُّب كل من هذه الحدود وتطبّق في الأقسام المختلفة لنظام التشغيل.

وبالعكس، يوفّر برنامج إدارة الكائنات NT برنامجاً خدماتياً مركزياً لحساب الموارد. ويعيّن لكل مستعمل حصّة تحدّد كميّة ذاكرة النظام التي تستطيع المعالجات إستعمالها جماعياً. وبشكل مشابه، تحتوي على ترويسة كائن على صفة تسمى حسابات الحصص تحدّد ما يطرحه برنامج إدارة الكائنات من الحصص المخصّصة للمعالجة عندما تفتح شعبة في المعالجة مقبض إلى الكائن. تستطيع شعَب المعالجة فتح العديد من المقابض حيث يطرح برنامج إدارة الكائنات حصتها في كل مرة. وإذا فتحت المعالجات العديد من المقابض واستهلكت حصة المستعمل، عندئذٍ يجب أن تغلق شعبها بعض مقابض الكائن قبل أن تفتح المزيد منها. وهكذا، يحدُّ برنامج إدارة الكائنات من إستعمال المعالجة (وبالتالي المستعمل) للموارد عن طريق مراقبة كمية الذاكرة المشغولة من قبَل الكائنات التي فتحت إليها مقابضاً من قبَل المعالجة. (إضافة إلى حصص الكائنات، يفرض برنامج إدارة الكائنات NT حصة على كميّة وقت المعالج التي يمكن أن تستعملها كل معالجات المستعمل).

3-2-3 طرُق الكائن:

يستخدم برنامج إدارة الكائنات أوجه تشابه الكائنات لكي يستطيع إدارتها بشكل متناسق. لكن الكائنات تتّصف بأوجه إختلاف أيضاً، وبعض هذه الإختلافات واضح. ويجب

على برنامج إدارة الكائنات أن يكون أكبر بكثير وأكثر تعقيداً لكي يتمكّن من إستيعاب خصوصيّات كل أنواع الكائنات المختلفة. ويجب أن يتغيّر إذا أُضيف في المستقبل نوع كائن جديد إلى النظام. ولمنع ذلك، يزوّد برنامج إدارة الكائنات عقيفات تستعملها مكوّنات البرنامج التنفيذي NT الأخرى لتنفيذ المهام الخاصة بأنواع الكائنات. تسمّى هذه العقيفات طرُق الكائن.

عندما ينشىء مكوّن تنفيذي نوع كائن جديد، فإنه يستطيع تسجيل طريقة واحدة أو أكثر بواسطة برنامج إدارة الكائنات. بعدها، يستدعي برنامج إدارة الكائنات الطرُق عند نقاط محدّدة خلال مدة إستعمال هذا النوع من الكائنات، عادة عند إنشاء كائن أو حذفه أو تعديله. تسرد الطرق التي يدعمها برنامج إدارة الكائنات في الجدول (5-3).

إحدى أمثلة إستعمال طريقة الإغلاق تتمّ في نظم الدخل / الخرج. فبرنامج إدارة الدخل / الخرج يسجّل طريقة إغلاق لنوع كائن الملفّ ويستدعي برنامج إدارة الكائنات طريقة الإغلاق في كل مرة يغلق فيها مقبض كائن ملفّ. وتدقّق طريقة الإغلاق لجهة إحتواء المعالجة التي تغلق مقبض الملفّ على أقفال ظاهرة على الملفّ، حيث تزيلها إذا وُجدت. إن عمليّة البحث عن أقفال الملفّ ليست عمليّة متوجّبة على برنامج إدارة الكائنات.

يستدعي برنامج إدارة الكائنات طريقة حذف، إذا سجلت، قبل حذف كائن مؤقّت من الذاكرة. فمثلاً، يسجّل برنامج إدارة الذاكرة الظاهريّة (VM) طريقة حذف لنوع كائن المقطع الذي يخلي الصفحات الفعليّة المستعملة من قبَل المقطع. ويتأكد أيضاً من حذف أية بنيات

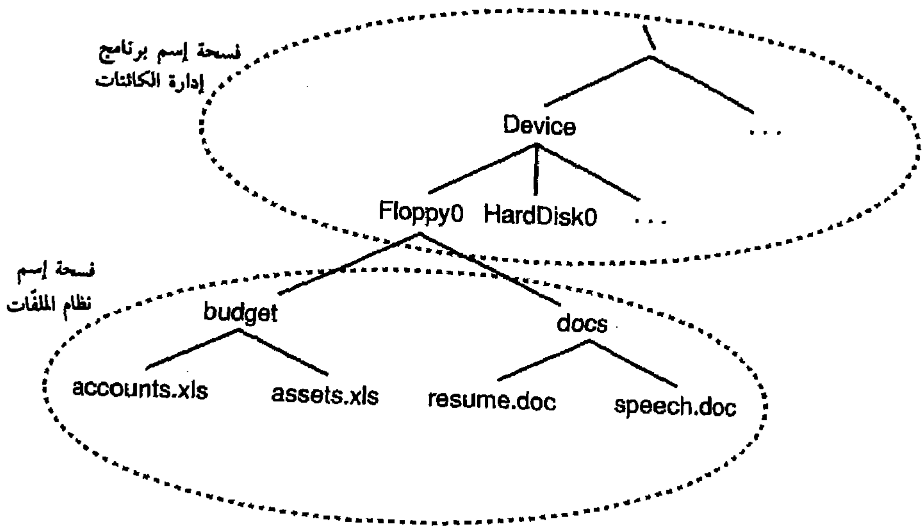
الطريقة	متى تستدعي الطريقة
فتح	عند فتح مقبض كائن
إغلاق	عند إغلاق مقبض كائن
حذف	قبل حذف كائن من قِبَل برنامج إدارة الكائنات
الإستعلام عن إسم	عندما تطلب شعبة إسم كائناً، مثل ملف موجود في حقل كائن ثانوي
إعراب	عندما يبحث مدير الكائنات عن إسم كائن موجود في نطاق كائن ثانوي.
الأمان	عندما تقرأ المعالجة أو تغيّر الحماية لكائن، مثل ملفّ موجود في حقل كائن ثانوي

الجدول (5-3)

طرق الكائن

بيانات داخليّة حدّدها برنامج إدارة VM لمقطع وذلك قبل حذف كائن المقطع. وهذا ما لا يستطيع برنامج إدارة الكائنات القيام به لأنه لا يعرف وغير متعلّق بالأعمال الداخليّة لبرنامج إدارة VM. تنفذ طرق الحذف لأنواع الكائنات الأخرى نفس الوظائف.

تتيح طريقة التحليل اللغوي (وبشكل مشابه، طريقة الإستعلام عن إسم) لبرنامج إدارة الكائنات، التحكّم بإيجاد كائن لبرنامج إدارة كائنات ثانوي. ويجد برنامج إدارة الكائنات الثانوي كائناً موجوداً خارج فسحة إسم برنامج إدارة الكائنات في حقل كائن مختلف. إن نظام الدخل / الخرج هو خير مثال على ذلك. راجع الشكل الذي عُرِض سابقاً.



إن الكائن Floppy0 هو كائن جهاز، من نوع خاص معرّف ومستعمل من قِبَل نظام الدخل / الخرج. وفي فسحة إسم برنامج إدارة الكائنات، يمثّل كائن الجهاز نقطة الإنطلاق نحو حقل كائن نظام الملفّات، وهو ما يجهله برنامج إدارة الكائنات.

عندما قام نظام الدخل / الخرج بإنشاء نوع كائن الجهاز، سجّل له طريقة تحليل لغوي. وهكذا وعندما يبحث برنامج إدارة الكائنات عن إسم كائن، فإنه يعلّق بحثه عندما يواجه كائناً في مسار تتعلّق به طريقة التحليل اللغوي. ويستدعي برنامج إدارة الكائنات طريقة التحليل اللغوي حيث يمرّرها إلى بقيّة إسم الكائن الذي يبحث عنه.

فمثلاً، عندما تفتح المعالجة مقبضاً إلى كائن يسمّى \Device\Floppy0\

\docs\resume.doc يستعرض برنامج إدارة الكائنات شجرة إسمه إلى أن يبلغ كائن الجهاز المسمى Floppy0. وعندما يشاهد طريقة تحليل لغوي متعلّقة بهذا الكائن، يقوم باستدعاء الطريقة ويمرّر إليها بقيّة إسم الكائن الذي يبحث عنه ـ وفي هذه الحالة، \docs\resume.doc. فإن طريقة التحليل اللغوي لكائنات الجهاز هي روتين دخل / خرج. يستلم الروتين نضيد الإسم ويمرّره إلى نظام الملفّات المناسب الذي يحدّد موقع الملفّ المناسب ويفتحه.

كذلك، تترجم كائنات الربط الرمزي التي وصفت في القسم 3-2-1-3، بواسطة طريقة التحليل اللغوي. فنوع كائن الربط الرمزي يتّصف بطريقة تحليل لغوي متعلّقة به. وتأخذ الطريقة الإسم وتستبدله بإسم آخر، ثم تستدعي برنامج إدارة الكائنات ليعاود بحثه عن الكائن. (وإذا احتوى الإسم الجديد على إسم كائن ربط رمزي أيضاً، تستدعي مجدداً طريقة التحليل اللغوي).

تشابه طريقة الأمان المستعملة من قِبَل نظام الدخل / الخرج، طريقة التحليل اللغوي. ويتم إستدعاءها عند محاولة شعبة تغيير معلومات الأمان التي تحمي الملفّ. وتختلف هذه المعلومات للملفات عن تلك العائدة للكائنات الأخرى لأن معلومات الأمان مخزّنة في الملفّ نفسه عِوَضاً عن تخزينها في الذاكرة. لذلك، يجب إستدعاء نظام الدخل/ الخرج لإيجاد معلومات الأمان وتغييرها.

3-3 حماية الكائنات:

رغم كون تسمية موارد النظام ومشاركتها وحسابها أسباب جيّدة لكي يستعمل البرنامج التنفيذي NT نموذج كائن، غير أن أكثر الأسباب أهمية هي ضمان كون النظام Windows NT نظام تشغيل آمن.

إن أمان نظام التشغيل هو من أهم المواضيع. يجب على المستعملين للنظام الآمن المتعدّد حماية ملفّات المستعمل وذاكرته وموارده الأخرى من المستعملين الآخرين. ويجب عليه حماية بيانات نظام التشغيل وملفاته وذاكرته هي برامج المستعمل ويجب عليه مراقبة محاولات تجاوز مزايا الأمان الخاصة به وما شابه. لقد عرّفت وزارة الدفاع الأميركيّة مزايا نظام تشغيل التي تجعله آمناً. تصنّف هذه المزايا في سبعة مستويات أمان وكل مستوى منها أكثر تشدُّداً من السابق.

عند المستوى Class C2، وهو الهدف الأولي للنظام Windows NT، يجب أن تتواجد المزايا التالية:

- البرنامج الخدماتي الآمن للتسجيل، يطلب من المستعملين تعريف أنفسهم عن طريق إدخال معرّف تسجيل خاص وكلمة سرّ قبل السماح لهم بالوصول إلى النظام.

- التحكُّم بالوصول الإستنسابي، يتيح لمالك مورد تحديد المستعملين الذين يستطيعون الوصول إلى المورد وما يمكن أن يقوموا به عليه. ويقوم المالك بذلك عن طريق منح حقوق الوصول إلى مستعمل أو إلى مجموعة من المستعملين.

- التدقيق، يوفّر القدرة على كشف الأحداث المهمة المتعلّقة بالأمان وتسجيلها أو أية محاولة لإنشاء موارد نظام أو حذفها. وهو يستعمل معرّفات التسجيل لتسجيل هوية المستعمل.

- حماية الذاكرة، تمنع أي مستعمل من قراءة المعلومات المكتوبة من قبل مستعمل آخر بعد إرجاع بنية بيانات إلى نظام التشغيل. ويعاد تحفيز الذاكرة بعد إعادة إستعمالها.

لن تحتاج كل برامج تركيب النظام Windows NT لكل آليات الأمان التي يوفّرها النظام. لذلك، يتيح نظام الأمان لمدير النظام تنسيق تتابع التسجيل، على سبيل المثال، أو تعديل تجميع المعلومات وكمّيتها في سجل تدقيق.

إن المرافق الحسّاسة جداً للنواحي الأمنيّة، مثل الإنشاءات العسكرية، تتطلّب مستوى أمان أعلى مما يوفّره النظام Windows NT مبدئياً. لذلك، صمّم النظام Windows NT حول مستوى الأمان Class B2 وهو مستوى معروف بإسم التحكُّم بالوصول الإجباري (Mandatory Access Control)، حيث يعينّ لكل مستعمل مستوى تصريح آمن ويمنع من توفير الوصول إلى الموارد المحميّة للمستعملين بمستوى أدنى. فمثلاً، في مرافق الحكومة الأميركيّة الآمنة، قد يحتوي مستعمل على تصريح أمان «سرّي» وتصريح أمان «سرّي جداً» لمستعمل آخر. يضمن التحكُّم بالوصول الإجباري للمستعمل الذي يتّصف بمستوى أمان «سرّي جداً»، منع المستعمل السابق من الوصول إلى أية معلومات «سريّة جداً»، حتى إذا إستعمل التحكُّم بالوصول الإستنسابي. وبشكل مشابه، يتطلّب مستوى الأمان B2 معرفة «الأقسام»، أي فصل مجموعات المستعملين عن بعضها البعض. يفيد هذا النوع من الحماية في الصناعات مثل البورصة أو عمليّات الدمج إلى تعارض المصالح.

إن نظام الأمان في Windows NT متعدّد الأوجه، لكن حماية الكائنات هو جوهر التحكُّم بالوصول الإستنسابي والتدقيق (ولاحقاً، التحكُّم بالوصول الإجباري). إن فكرة نظام الأمان في Windows NT هي إنشاء بوّابة يجب على كل مستعمل لنظام الموارد أن يدخل منها. ولأن كل موارد النظام الممكن تسويتها مستخدمة ككائنات، ويصبح برنامج إدارة الكائنات NT البوّابة.

إن الأقسام الفرعيّة التالية تشرح حماية الكائن من ناحيتين: الأول، التحقُّق من هوية المستعملين والثانية التحكُّم بمن يستطيع من المستعملين الوصول إلى كائنات محدّدة.

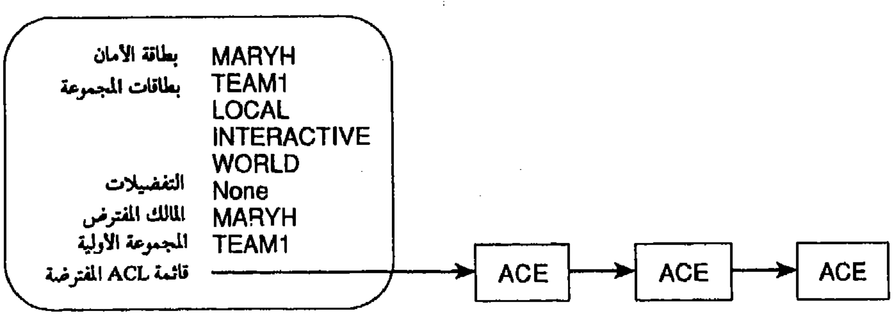
1-3-3 صفات الوصول:

للتحكُّم بمن يستطيع مناولة الكائنات، يجب أن يتأكّد نظام الأمان من هويّة كل مستعمل. لذلك، فإن أول سطر حماية في النظام Windows NT هو المتطلبات التي يجب أن يدخل كل مستعمل على النظام.

كما شرح الفصل الثاني «نظرة شاملة حول النظام»، فالنظام الفرعي للحماية المتكاملة، أي النظام الفرعي للأمان، مسؤول عن إثبات أصالة المستعملين ـ أي، التحقُّق من مطابقة المعلومات التي يسجّلها المستعمل مع تلك الموجودة في قاعدة بيانات الأمان. بعد أن يثبت النظام الفرعي للأمان أصالة معلومات التسجيل، فإنه ينشىء كائناً يلحق بشكل دائم بمعالجة المستعمل. يسمى هذا الكائن صفة الوصول، وهو يستعمل كهويّة رسميّة للمعالجة عند محاولته إستعمال موارد نظام. تظهر صفة وصول عيّنة في الشكل (3-8).

الصفة الأولى في هذا المثال هي بطاقة الأمان الشخصيّة للمستعمل. وهي معرّف يتوافق عادة مع إسم تسجيل المستعمل. ومن الإنشاءات الكبيرة، قد تشمل بطاقة الأمان إسم قسم أو فرع المستعمل (مثلاً ENGINEERING_MARYH). تشكّل بطاقات الأمان لمجموعة من قوائم بطاقات المستعملين. أما الصفة الثانية المبيّنة في الشكل (3-8) فهي قائمة بالمجموعات التي تنتمي إليها MARYH. يعرّف النظام Windows NT عدّة معرّفات مجموعات قياسيّة شموليّة في صفة MARYH.

عندما تحاول معالجة فتح مقبض إلى كائن، يستدعي برنامج إدارة الكائنات مراقب مراجع



الشكل (3-8)
صفة الوصول العيّنة

الأمان. يجلب مراقب مراجع الأمان الصفة المتعلقة بالمعالجة وإستعمال بطاقة الأمان الخاصّة بها وقائمة المجموعات لتحديد إمكانيّة وصول المعالجة إلى الكائن.

كذلك تتمُّ حماية مجموعة صغيرة من خدمات النظام الحسّاس للأمان (مثل صفة الإنشاء) من الإستعمال. وتسرد صفة التفضيلات أياً من هذه الخدمات الخاصة التي يمكن أن يستعملها المستعمل. معظم المستعملين لا يتّصفون بتفضيلات.

يصبح المستعمل الذي ينشىء كائن مالكه ويستطيع تحديد من يستطيع إستعماله. إن صفة قائمة التحكُّم بالوصول (ACL) المفترضة لصفة الوصول هي قائمة أوليّة للحماية المطبّقة على الكائنات التي أنشأها المستعمل. توفّر صفة المجموعة الأوليّة القدرة على تجميع بطاقات الأمان في مجموعات لأغراض تنظيميّة وهي مزيّة تتمتّع بها معظم أنظمة التشغيل، بما فيها POSIX.

تشرح تفاصيل بطاقات الأمان وقائمة ACL في القسم التالي. راجع الشكل (3-9) الذي يلخّص الصفات والخدمات المطبّقة على كائنات صفة الوصول.

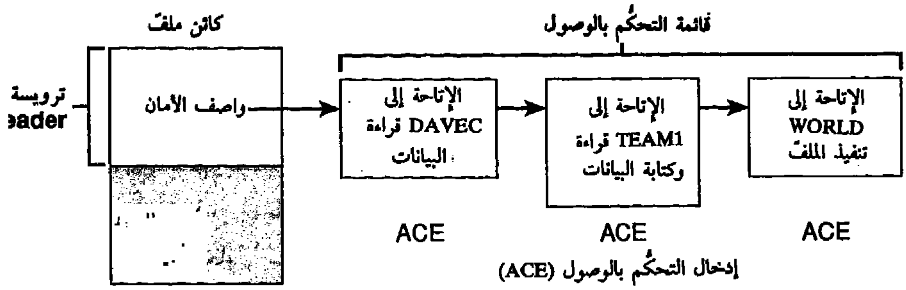
إضافة لخدمات الإنشاء والفتح والإستعلام، يظهر أيضاً خدمة ضبط الصفة. ضبط الصفات في كائن هي خدمة عامة متوفّرة للعديد من كائنات البرنامج التنفيذي NT. والخدمات الثلاث المتبقّية مستعملة مبدئياً من قِبَل براجيّات إدارة الأمان.



الشكل (3-9)
كائن صفة الوصول

2-2-3 قوائم التحكُّم بالوصول:

يتمُّ تعيين واصف أمان لكل الكائنات بما فيها الملفّات والشعَب والأحداث وصفات الوصول وذلك عند إنشائها. إن المزيّة الرئيسيّة لواصف الأمان هي قائمة الحماية المطبقة على الكائن والتي تسمى قائمة التحكُّم بالوصول (ACL). ويحتوي مالك الكائن وهو الشخص الذي ينشئه عادة، على تحكُّم إستنسابي بالوصول على الكائن ويستطيع تغيير قائمة ACL للكائن لإتاحة الوصول إلى الكائن أو لعدم إتاحة الوصول إليه. الشكل (3-10) هو مثال مبسّط عن كائن ملفّ وقائمة ACL.



الشكل (3-10)
قائمة التحكُّم بالوصول (ACL)

يعرف كل إدخال في ACL كإدخال تحكُّم بالوصول (ACE). يحتوي إدخال ACE على بطاقة أمان ومجموعة من حقوق الوصول، وقد يتاح لمستعمل ذات بطاقة أمان مطابقة بحقوق الوصول المسردة أو حجبها عنه أو إتاحتها مع تدقيق. يشكّل تراكم حقوق الوصول الممنوحة من قِبَل إدخال ACE الإفرادي مجموعة حقوق الوصول الممنوحة من قِبَل ACL.

إفترض إنك تحاول سرد ملفّ. إذا كانت القائمة ACL لكائن الملفّ تحتوي الإدخال ACE يحتوي حق الوصول المسمى قراءة البيانات، فإنه يُتاح لك سرد الملفّ. إضافة لذلك، إذا كانت العمليّة التي تحاول القيام بها هي عمليّة بتفضيلات مثل صفة الإنشاء، يجب أن تتّصف بتفضيلات لإنشاء صفة وصول. وإلا يمنع الوصول كما يظهر في الشكل (3-10)، يمكن إنشاء الإدخال ACE لبطاقات أمان لمجموعة. يتّصف DAVEC بالوصول إلى القراءة لكائن الملفّ ويتّصف أفراد المجموعة TEAM1 بالوصول إلى القراءة والكتابة ويتّصف كل المستعملين الآخرين بالوصول للتنفيذ.

لتحديد القائمة ACL التي يجب أن تعيّن لكائن جديد، يطبّق نظام الأمان ثلاث قواعد منع تبادليّة في الترتيب التالي:

1 ـ إذا وفّر مستدعي قائمة ACL عند إنشاء الكائن، يطبّق نظام قائمة ACL هذه على الكائن.

2 ـ إذا لم يزوّد مستدعي قائمة ACL وكان الكائن يتّصف بإسم، يبحث نظام الأمن في القائمة ACL على دليل الكائن حيث يخزّن إسم الكائن الجديد. وقد تعلّم بعض إدخالات ACE لدليل الكائن بالعلامة «تأصيل»، وهذا يعني أنه يجب تطبيقها على الكائنات الجديدة المنشأة في دليل الكائن. وإذا تواجدت أي من إدخالات ACE القابلة للتأصّل، يشكّلها نظام الأمان في قائمة ACL والتي تلحق بالكائن الجديد.

3 ـ إذا لم يحدث أي من الحالتين الأوّلتين، يستردّ نظام الأمان قائمة ACL المفترضة من صفة الوصول للمستدعي ويطبّقها على الكائن الجديد.

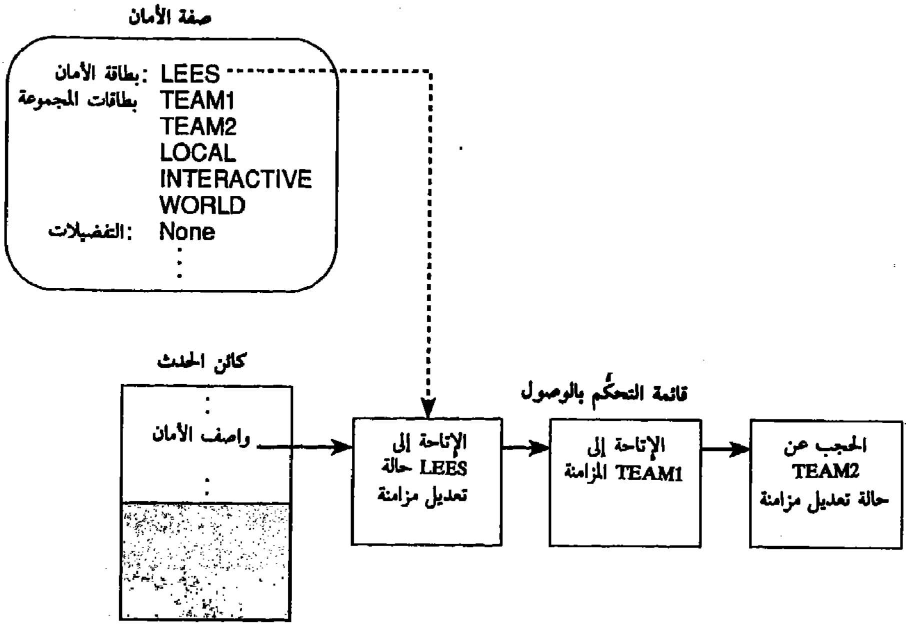
إضافة إلى قائمة ACL، يحتوي واصف الأمان لكائن على مجال ينظم التدقيق في الكائن. والتدقيق هي قدرة نظام الأمان على «التجسّس» على كائنات منتقاة ومستعمليها ولتوليد الرسائل أو الإنذارات عندما يحاول شخص ما عمليّة محظورة على كائن. فمثلاً، يستطيع نظام الأمان التدقيق بمحاولات قراءة ملفّ نظام أو تعديله. وإذا حاول شخص ما تغيير الملفّ، يكتب نظام الأمان رسالة إلى سجلّ التدقيق يعرّف المستعمل بواسطة بطاقة الأمان. يستطيع برنامج إدارة النظام إصدار تقارير أمان تجمع معلومات من السجل. ولأنظمة الأمان المرتفعة، صمّم نظام الأمان ليصدر إنذاراً سمعياً أو بصرياً على ماكنة مدير الأمان عندما يحصل الفعل. يساعد التدقيق على تخصيص مخاطر العبث بالحاسوب.

3-3-3 الإستنتاج:

تعرّف صفة الوصول معالجة (وشعبها) إلى نظام التشغيل، بينما يسرد واصف الأمان أياً من هذه المعالجات (أو مجموعات المعالجات) تستطيع الوصول إلى كائن. وعندما تفتح شعبة مقبض إلى كائن، يضع برنامج إدارة الكائنات ونظام الأمان هذه المعلومات سويّة ليحدّد تزويد المستدعي أو عدم تزويده المقبض الذي يطلبه.

يوضّح الشكل (3-11) على الصفحة التالية ماذا يحدث عندما يفتح المستعمل LEES مقبضاً، حيث يطلب وصول متزامن إلى كائن الحدث.

عند التدقيق في قائمة ACL، يدقّق نظام الأمان في القائمة من أول إدخال ACE إلى



الشكل (11-3)
التدقيق في حماية كائن

الأخير. وعندما يجد بطاقة الأمان أو بطاقة المجموعة للمستدعي، فإنه يوقف بحثه ويدقّق لجهة إتاحة الإدخال ACE لنوع الوصول الذي يحاول أن يقوم به المستعمل. فإذا وجد ACE يتيح ذلك، فإنه يوقف بحثه ويرجع مقبض إلى المستدعي. وإذا بلغ نهاية القائمة دون إيجاد بطاقة الأمان أو المجموعة للمستدعي، يرفض طلب المستدعي.

في الشكل (11-3)، تتيح القائمة ACL الخاصة بكائن الحدث للمستعمل LEES الوصول إلى المزامنة في إدخالها الأول. ولأن LEES تطلب الوصول إلى المزامنة، يوقف نظام الأمان بحثه فوراً ويرجع برنامج إدارة الكائنات مقبض إلى LEES. الذي يحتوي على وصول المزامنة إلى الحدث. لاحظ أن الإدخال ACE الثالث يحجب عن LEES الوصول للمزامنة وفقاً لعضويتها في TEAM 2. لكن، بسبب ترتيب إدخالات ACE في قائمة الحكم بالوصول هذه، يتمّ تجاهل إدخال ACE الثالث في هذه الحالة. (هذا مثال إصطناعي لأن النظام يضع عادة إدخالات ACE التي تحجب الوصول في بداية القائمة).

لن تكون عملية التدقيق هذه التي يقوم بها نظام الأمان في كل مرة تستعمل فيها معالجة مقبض، عملية كافية. فقائمة ACL تحتوي عدّة إدخالات، ويمكن لمعالجة الوصول إلى عدّة

كائنات خلال فترة خدمتها، ويمكن أن تتواجد عدّة معالجات نشطة في نفس الوقت. لذلك، يتمُّ التدقيق فقط عند فتح مقبض، وليس في كل مرة يستعمل فيها المقبض. (لاحظ أنه بما أن شيفرة نمط النواة تستعمل المؤشرات بدلاً من المقابض للوصول إلى الكائنات، لا ينفذ التدقيق بالوصول عندما يستعمل نظام التشغيل الكائنات. بمعنى آخر، يثق البرنامج التنفيذي NT بناحية الأمان الخاصة به).

وفي المرة التالية التي تستعمل فيها LEES مقبض الحدث، يقارن برنامج إدارة الكائنات الوصول الممنوح (المزامنة) المخزّن في المقبض مع نوع الوصول المطبّق بواسطة الخدمة المستدعاة. وإذا استدعت خدمة إنتظار، ينجح الإستدعاء. وإذا إستدعت ضبط الحدث، تخفق الخدمة. ولاستدعاء ضبط الحدث، يجب عليها إما فتح مقبض جديد وطلب الوصول لتعديل الحالة.

لاحظ أنه بعد فتح مقبض بنجاح من قبَل معالجة، لا يمكن إلغاء حقوق الوصول الممنوحة من قبَل نظام الأمان حتى إذا تغيّرت القائمة ACL للكائن. فالمقبض القديم تأصّل لأن المطوِّرين حدّدوا أن التدقيق في الأمان الكافي أكثر أهميّة من القدرة على إلغاء حقوق الوصول الممنوحة. وهذه القدرة الأخيرة تتطلّب تدقيقاً كاملاً بالأمان في كل مرة يستعمل فيها مقبض عوَضاً عن التدقيق فقط عند إنشاء المقبض كما يحدّد ذلك التصميم الحالي. إن تحسين الأداء الواضح بواسطة تخزين حقوق الوصول الممنوحة مباشرة في المقابض هو مهمّ وخاصة للكائنات ذات قوائم ACL الطويلة.

4-3 في الختام:

تمثّل كائنات البرنامج التنفيذي NT مفهوماً موحّداً في النظام Windows NT. وهي توفّر أُسُس إدارة موارد النظام بشكل متناسق. وهي تخدم أيضاً كنقطة تركيز للمهام المهمّة مثل تسمية الموارد ومشاركتها وحمايتها. إضافة لذلك، فهي تزوّد مجموعة من الأُسس التي تستعملها الأنظمة الفرعيّة للمحيط لتطبيق إصدارات الكائنات الخاصة بها والموارد المشابهة للكائنات. يستعمل كل نظام فرعي لمحيط كائنات تنفيذيّة لتوفير البرامج الخدماتيّة والموارد التي تتوقّعها تطبيقات المستضاف.

تعتمد كائنات نمط المستعمل الممثّلة في هذا الفصل على مجموعة كائنات أوليّة مستخدمة من قبَل النواة NT. تشعُّ كائنات النواة وقدراتها في الفصل السابع، «النواة». وفي الفصل التالي، سيتمُّ شرح كائنين خاصّين متكاملين مع وظائف Windows NT: المعالجات والشعَب.

4

المعالجات والشعب

في الإصدارات السابقة للنظام MS-DOS، كان المستعمل يُشغِل برنامجاً واحداً فقط في كل مرة. وكان يشغّل برنامجاً واحداً وينتظر أن ينتهي قبل أن يشغّل برنامجاً آخر. لكن بـظلّ Windows ، يستطيع المستعمل تشغيل أكثر من برنامج واحد في كل مرة أو حتى عدّة نسخ من نفس البرنامج في نفس الوقت. يبرز هذا التغيير إختلافاً دقيقاً مهمّاً لهذا الفصل: الفرق بين برنامج ومعالجة. فالبرنامج هو تتابُع إستاتي للتعليمات، بينما المعالجة هي الإنفاذ الدينامي لبرنامج سويّة مع موارد النظام المطلوبة لتشغيل البرنامج.

تمثّل المعالجة وحدة ملكيّة موارد والعمل الواجب تنفيذه. وهي وسيلة نظام التشغيل لتنظيم العديد من المهام التي تنفّذها. يحدّد نظام التشغيل قسماً من موارد الحاسوب إلى كل معالجة ويضمن إرسال كل برنامج معالجة للتنفيذ في طريقة مرتّبة وزمنيّة.

تتّصف عادة أنظمة التشغيل بجسم من الشيفرات التي تدير إنشاء المعالجات وحذفها والعلاقات بين المعالجات. تسمى هذه الشيفرة بنية المعالجة وتستخدم في النظام Windows NT من قبَل برنامج إدارة المعالجة. صمّم Mark Lucovsky، مطوِّر Windows NT والذي كتب مكوّنات بنية المعالجة لنظام UNIX ولنظام تشغيل كائني، وكتب برنامج إدارة المعالجة للبرنامج التنفيذي NT. وقد عرّف أهدافها الأساسيّة في جملة واحدة: لتوفير مجموعة من خدمات المعالجة المحليّة التي تستطيع الأنظمة الفرعيّة للمحيط إستعمالها لمحاكاة بنيات المعالجة الفريدة. وقد تطوّر هذا الهدف، هدف Windows NT في توفير محيطات نظام تشغيل متعدّدة تعمل في نمط المستعمل.

تستخدم أنظمة تشغيل مختلفة المعالجات في طرق مختلفة. والمعالجات تتغيّر في كيفيّة عرضها (بنيّة بياناتها) وكيفيّة تسميتها وكيفيّة حمايتها والعلاقات المتواجدة بينها.

تحتوي معالجات NT المحليّة عدّة خصائص تختلف عن المعالجات في أنظمة التشغيل الأخرى:

■ تستخدم معالجات NT ككائنات، ويتمّ الوصول إليها باستعمال خدمات للكائن.

■ يمكن أن تحتوي معالجة NT عدّة شعَب تنفّذ ضمن فسحة عنوانها.

■ تحتوي كائنات المعالجة وكائنات الشعبة قدرات مزامنة داخلية.

■ لا يحافظ برنامج إدارة المعالجة NT على علاقة أم / تابع أو علاقات أخرى ضمن المعالجات التي تنشئها.

يشرح هذا الفصل طبيعة المعالجات بشكل عام وبنية معالجات البرنامج التنفيذي NT بشكل خاص. وهو يبدأ بتعريف المعالجة ثم يشرح كيف يستخدم برنامج إدارة المعالجة NT لإصدار المعالجة. ثم يتبع ذلك مقدّمة حول الشعَب بما فيها شرح حول الحاجة للشعَب والمصطلحات المتعلّقة بالشعَب وكيف يستخدمها برنامج إدارة المعالجة NT وإصدار المعالجات التي توفّرها الأنظمة الفرعيّة المحيط NT إلى البرامج التطبيقيّة.

1-4 ما هي المعالجة؟

في أعلى مستوى تجريدي، تتألف المعالجة في التالي:

■ برنامج قابل للتنفيذ، يعرّف الشيفرة والبيانات الأوليّة.

■ فسحة عنوان خاصة، وهي مجموعة من عناوين الذاكرة الظاهريّة التي تستطيع المعالجة إستعمالها.

■ موارد النظام، مثل الإعلام الإشاري ومنافذ الإتصال والملفّات التي يحدّدها نظام التشغيل إلى المعالجة عند تنفيذ البرنامج.

في النظام Windows NT، يجب أن تتضمّن المعالجة عنصراً رابعاً قبل القيام بأي شيء:

■ على الأقلّ شعبة تنفيذ واحدة.

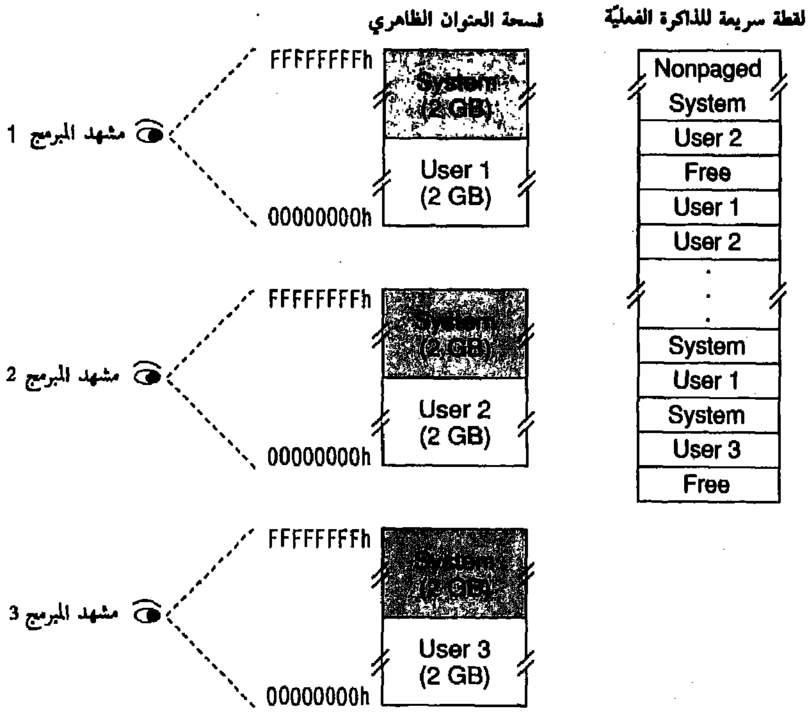
الشعبة هي الوحدة المستقلّة ضمن معالجة تجدولها النواة NT للتنفيذ. ودونها لا يمكن تشغيل برنامج المعالجة.

تشرح الأقسام الفرعيّة التالية المعالجات بتفصيل أكبر، حيث تعاين أولاً فسحة عنوان المعالجة؟ مواردها. ويعرض القسم الفرعي اللاحق موضوع الشعَب.

1-1-4 فسحة العنوان:

يوصي المنطق العام إلى ضرورة منع معالجة واحدة من إستخدام تحكُّم غير محدود على المعالجات الأخرى. وإستعمال نظام الذاكرة الظاهريّة هو أحد الطرق التي يعتمدها النظام Windows NT. وبواسطة الذاكرة الظاهريّة، يحتوي المبرمجين (والمعالجات التي يُنشئونها) على مشهد منطقي للذاكرة التي تتوافق وتصميمها الفعلي. راجع الشكل (1-4).

وفي كل مرة تستعمل المعالجة عنوان ذاكرة، يترجم نظام الذاكرة الظاهريّة العنوان في الذاكرة الفعليّة. وهو يمنع المعالجات من الوصول مباشرة إلى الذاكرة الظاهريّة المشغولة من قبَل المعالجات الأخرى أو بواسطة نظام التشغيل. لتنفيذ شيفرة نظام التشغيل أو الوصول إلى ذاكرة نظام التشغيل، يجب تشغيل شعبة في نمط المعالج الحرّ المسمّى نمط النواة. لكن معظم المعالجات هي في نمط المستعمل ـ أي المعالجات التي تشتغل شعَبِها مبدئياً في نمط المعالج المقيّد المسمّى نمط المستعمل.



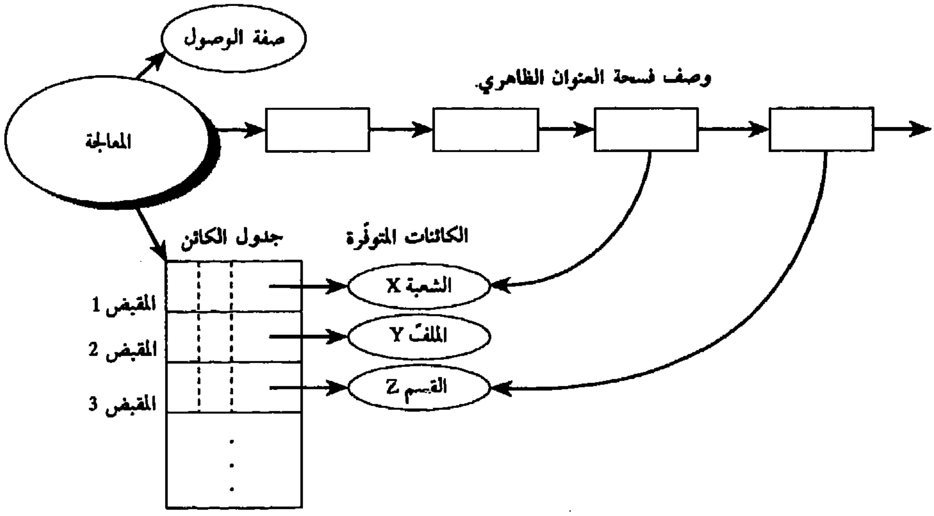
الشكل (1-4)
الذاكرة الظاهريّة مقابل الذاكرة الفعليّة

تكتسب شعبة نمط المستعمل الوصول إلى نظام التشغيل باستدعاء خدمة نظام. وعندما تستدعي الشعبة الخدمة، يحجزها المعالِج ويحوِّل تنفيذها من نمط المستعمل إلى نمط النواة. ويتحكّم نظام التشغيل بالشعبة بحيث يجعل الإيعازات المستقلّة التي تمرّرها الشعبة إلى خدمة النظام صالحة ثم تنفّذ الخدمة. يعاود نظام التشغيل تحويل الشعبة إلى نمط المستعمل قبل إرجاع التحكُّم إلى برنامج المستعمل. بهذه الطريقة، يحمي نظام التشغيل نفسه وبياناته من الملاحقة والتعديل من قبَل معالجات المستعمل.

يركِّز هذا الفصل على المعالجات في نمط المستعمل والتي تمثِّل أكثريّة المعالجات في النظام Windows NT في أي وقت. تشتغل البرامج التطبيقيّة في نمط المستعمل، وكذلك الأنظمة الفرعيّة المحميّة للنظام Windows NT. وهذه الأخيرة معالجات ملقِّم في نمط المستعمل توفِّر قدرات هامة لنظام التشغيل. وهي تستخدم كملقّمات لتبسيط نظام التشغيل المرجعي وجعله مدوداً. تشتغل الأنظمة الفرعيّة في نمط المستعمل بحيث تتمّ حماية كل فسحة عنوان واحد في معالجات التطبيق. ومن الأنظمة الفرعيّة الأخرى (راجع الفصل الخامس لمزيد من المعلومات).

2-1-4 تجميع الموارد:

إضافة لفسحة العنوان الخاصة، تحتوي كل معالجة على مجموعة من موارد النظام ملحقة بها. يظهر الشكل (2-4) معالجة نموذجيّة ومواردها.



الشكل (2-4)
معالجتها ومواردها

يوجد في أعلى المخطّط صفة الوصول للمعالجة والتي تمّ شرحُها في الفصل الثالث «برنامج إدارة الكائنات وأمان الكائن». لاحظ أن كائن الصفة يلحق مباشرة بالمعالجة بواسطة نظام التشغيل. وإذا أرادت المعالجة الحصول على معلومات إضافيّة حول صفة الوصول العائدة لها أو تغيير بعض صفاتها، يجب أن تفتح المعالجة مقبضاً إلى كائن الصفة. يحدّد نظام الأمان إمكانيّة القيام بذلك. وهذه المعالجة بالذات لم تتمكّن من فتح مقبض إلى صفة الوصول العائدة لها، وبالتالي لا يوجد سهم يمتدُّ من جدول الكائن إلى صفة الوصول.

تحت صفة الوصول، يوجد سلسلة من بنيات البيانات التي أنشأها برنامج إدارة الذاكرة الظاهريّة (VM) لتعقُّب العناوين الظاهريّة التي تستعملها المعالجة. ولا تستطيع المعالجة قراءة هذه البنيات أو تعديلها مباشرة. فبرنامج إدارة VM ينشئها ويعدّلها بطريقة غير مباشرة عندما يخصّص البرنامج الذاكرة. (يتمُّ شرح بنيات البيانات هذه في تفصيل أكبر في الفصل السادس «برنامج إدارة الذاكرة الظاهريّة»).

يظهر جدول كائن المعالجة في أسفل الشكل. ولقد فتحت المعالجة مقابض إلى أحد شعبها، إلى ملفّ وإلى مقطع من الذاكرة المشاركة. (يسجّل وصف العنوان الظاهري العناوين الظاهريّة المشغولة من قِبَل تكديس الشعبة وكائن المقطع، كما يُشار إلى ذلك بالأسهم من وصف العنوان الظاهري إلى هذه الكائنات).

إضافة إلى الموارد المبيّنة في الشكل، تحتوي كل معالجة على مجموعة من حدود حصص الموارد التي تقيّد كميّة الذاكرة التي تستطيع شعبها إستعمالها لفتح المقابض إلى الكائنات. كذلك، تحتوي كل معالجة على أولويّة تنفيذ مرجعيّة وصلة بمعالج مفترض، وهذه مواضيع سيتمُّ شرحها لاحقاً في هذا الفصل.

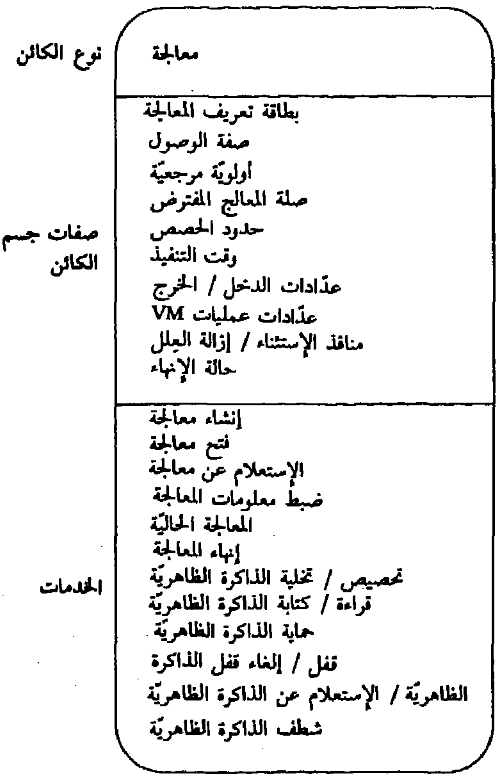
3-1-4 كائن المعالجة:

في البرنامج التنفيذي NT، المعالجات هي كائنات أُنشئت وحُذفت بواسطة برنامج إدارة الكائنات. يحتوي كائن المعالجة، كالكائنات الأخرى، على ترويسة ينشئها برنامج إدارة الكائنات ويحفزها. تخزن الترويسة صفات الكائن القياسيّة، مثل واصف الأمان لكائن المعالجة، وإسم المعالجة (إذا كانت تتّصف بواحد لأغراض المشاركة) ودليل الكائن حيث يخزّن إسمها.

يعرّف برنامج إدارة المعالجات الصفات المخزّنة في جسم كائنات المعالجة ويزوّد أيضاً خدمات النظام التي تستردُّ هذه الصفات وتغيّرها. توضح صفات وخدمات كائنات المعالجة في الشكل (3-4) على الصفحة التالية.

لاحظ أن جدول الكائن ووصف فسحة العنوان غير مُسرَدَين كجزء من كائن المعالجة. وهذا لأنه لا يمكن تعديلها مباشرة بواسطة المعالجات في نمط المستعمل، رغم كونها ملحقة بكائن المعالجة. يرسم الشكل فقط البيانات التي تستطيع شيفرة نمط المستعمل قراءتها أو ضبطها باستدعاء خدمات كائن المعالجة. يلخّص الجدول (4-1) على الصفحة التالية صفات كائن المعالجة.

تفرض العديد من صفات كائن المعالجة قيوداً على الشعَب المنفّذة داخل معالجة. فمثلاً، على حاسوب متعدّد المعالجات، قد تقيّد صلة المعالج شعَب المعالجة من التشغيل على مجموعة فرعيّة من المعالجات المتوفّرة وبشكل مشابه، تنظم حدود الحصص كميّة الذاكرة. وصفحات فسحة الملفّ ووقت التنفيذ التي تستعملها جماعياً شعَب المعالجة.



الشكل (4-3)
كائن المعالجة

الصفة	الغرض
بطاقة تعريف المعالجة	قيمة فريدة تعرّف المعالجة إلى نظام التشغيل.
صفة الوصول	كائن تنفيذي يحتوي معلومات الأمان المتعلّقة بالمستعمل المسجّل الممثّل بهذه المعالجة.
أولويّة مرجعيّة	أولويّة تنفيذ مرجعيّة لشُعَب المعالجة.
صلة المعالج المفترض	مجموعة المعالجات المفترضة حيث يمكن تشغيل شعَب المعالجة.
حدود الحصص	الكميّة القصوى لذاكرة النظام المعيّنة وغير المعيّنة لصفحات وتعيين صفحات لفسحة الملفّ ووقت المعالج الذي تستعمله معالجات المستعمل.
وقت التنفيذ	الكميّة الإجماليّة لوقت تنفيذ كل الشعَب في المعالجة.
عدّادات الدخل / الخرج	المتغيّرات التي تسجّل عدد ونوع عمليّات الدخل / الخرج التي نفّذتها شعَب المعالجة.
عدّادات عمليّات VM	المتغيّرات التي تسجّل عدد ونوع عمليّات الذاكرة الظاهريّة التي نفّذتها شعَب المعالجة.
منافذ الإستثناء / إزالة العلل	أقنية إتصال داخل المعالجة حيث يرسل برنامج إدارة المعالجة رسالة عندما تسبّب إحدى شعَب المعالجة إستثناء.
حالة الإنهاء	سبب إنهاء معالجة.

الجدول (4-1)

صفات كائن المعالجة

تساعد الأولويّة المرجعيّة المعالجة النواة NT على تنظيم أولويّة تنفيذ الشعَب في النظام. تختلف أولويّة الشعَب الإفراديّة لكنها تبقى دائماً ضمن مجال الأولويّة المرجعيّة لمعالجتها. تستطيع الأنظمة الفرعيّة للمحيط إستعمال الأولويّة المرجعيّة للكائن المعالجة لتحديدة شعب المعالجة المنتقاة أولاً من قبل النواة NT. فمثلاً يستدعي النظام Win 32 خدمات NT لرفع الأولوية المرجعية لمعالجة تطبيق أماميّة وتخفيض الأولويّة المرجعيّة لمعالجات تطبيق خلفيّة حيث تزوّد التطبيقات المتفاعلة دعم مقارنة مع الأخرى (راجع القسم 4-2-3). إن حدود الحصص وصلة المعالج والأولويّة المرجعيّة هي من ضمن صفات المعالجة وبنيات البيانات التي يمكن أنَ تتأهّل من معالجة واحدة إلى أخرى. يتمّ شرح تأصّل المعالجة في القسم 4-3-2-1.

إن منافذ إستثناءات وإزالة العِلل للمعالجة هي أقنية إتصال داخل المعالجة حيث يرسل نظام التشغيل رسائل عندما تولّد إحدى شعَب المعالجة إستثناء أمر عند إزالة العلل من معالجة

تنتظر شعبة في معالجة آخر عند المنفذ لتستلم الرسالة ولتتّخذ الفعل المناسب. فمثلاً، تستطيع شعبة نظام فرعي لمحيط «التنصُّت» عند منفّذ الإستثناء لإلتقاط الأخطاء المولّدة من قبَل معالجات المستضاف ويستطيع مزيل العلل إلتقاط الإستثناءات مثل نقاط فصل مزيل العلل. (راجع الفصل الخامس، «النظام Windows والأنظمة الفرعيّة المحميّة» لمزيد من المعلومات حول كائنات المنفّذ والأنظمة الفرعيّة للمحيط).

معظم خدمات كائن المعالجة واضحة. فخدمة إنشاء المعالجة هي مرنة وتتيح للأنظمة الفرعيّة المختلفة إنشاء معالجات بصفات أوليّة مختلفة. وتتيح خدمة المعالجة الحاليّة لمعالجة الحصول على مقبض بسرعة دون المرور عبر برنامج إدارة الكائنات. وتوقف خدمة إنهاء المعالجة شعَب المعالجة وتغلق أي مقابض كائن مفتوحة وتحذف فسحة العنوان الظاهري للمعالجة.

تستخدم خدمات الذاكرة الظاهريّة المبيّنة في الشكل (4-2) من قِبَل برنامج إدارة VM لكي يتطلّب كلٌ منها مقبض معالجة كبارامتر يُحدِّد للمعالجة الذاكرة الظاهريّة التي سيتمُّ الوصول إليها. يتمُّ وصف عمليات الذاكرة الظاهريّة في الفصل السادس «برنامج إدارة الذاكرة الظاهريّة».

2-4 ما هي الشعَب؟

إذا كنت تعرف موضوع الشعَب، قد تكون واجهت تعريفات مختلفة للشعَب بما فيها «وحدة تنفيذ» و «عدّاد برنامج مستقلّ» أو «وحدة مستقلّة مجدولة ضمن معالجة». ورغم كون كل من هذه التعريفات صحيحاً، لكنها ليست وافية بالكامل. فماذا يعني «وحدة تنفيذ»؟ وما هو الذي ينفّذ على معالج؟

فبينما تمثّل المعالجة منطقياً وظيفة يجب أن ينفّذها نظام التشغيل، تمثّل الشعبة إحدى المهام الفرعيّة المتعدّدة الممكنة والمطلوبة لتحقيق الوظيفة. فمثلاً، إفترض أن المستعمل بدأ تطبيق قاعدة بيانات في نافذة. يعرض نظام التشغيل قاعدة البيانات كمعالجة واحدة. وافترض أن المستعمل يطلب إنشاء تقرير عن جدول الرواتب من قاعدة البيانات وإرساله إلى ملفّ ـــ وهذه عمليّة طويلة. فخلال تنفيذ هذه العمليّة، يستطيع المستعمل إدخال إستعلام آخر عن قاعدة البيانات. يعرض نظام التشغيل كل طلب ـــ تقرير جدول الرواتب وقاعدة البيانات الجديدة ـــ كشعَب مستقلة ضمن معالجة قاعدة البيانات. ويمكن جدولة الشعَب للتنفيذ بشكل مستقلّ على المعالج، الذي يتيح تنفيذ المعالجتين في نفس الوقت (متزامنة). يوفّر نظام التشغيل الشعَب لتحقيق هذا التزامن في طريقة مناسبة وفعّالة يشرح مزيد حول هذا الموضوع لاحقاً.

إن ما يلي هي المكّونات الأساسيّة لشعبة في البرنامج التنفيذي NT.

- معرّف فريد يسمى بطاقة تعريف المستضاف.
- محتويات مجموعة مسجلات متطايرة تمثّل حالة المعالج.
- تكديسات، واحد ليُستعمل من قبَل الشعبة خلال التنفيذ في نمط المستعمل والآخر ليستعمل خلال التنفيذ في نمط النواة.
- منطقة تخزين خاصة لتستعمل من قبَل الأنظمة الفرعيّة ومكتبات وقت التشغيل ومكتبات الربط الديناميّ (DLLs).

تسمى المسجلات المتطايرة والتكديسات ومنطقة التخزين الخاصة سياق الشعبة. وتتغير البيانات الفعلية التي تؤلف سياق الشعبة من معالج إلى آخر.

تستقرُّ الشعبة ضمن فسحة العنوان الظاهري للمعالجة باستعمال فسحة العنوان للتخزين خلال تنفيذ الشعبة. وإذا وُجد أكثر من شعبة واحدة في نفس المعالجة، فإنها تشارك فسحة العنوان وكل موارد المعالجة بما فيها صفة الوصول والأولويّة المرجعيّة ومقابض الكائن في جدول الكائن. تجدول النواة NT الشعَب للتنفيذ على معالج. لذلك، يجب أن تكون محل معالجة NT شعبة واحدة قبل تنفيذها.

1-2-4 المهام المتعدّدة والمعالجة المتعدّدة:

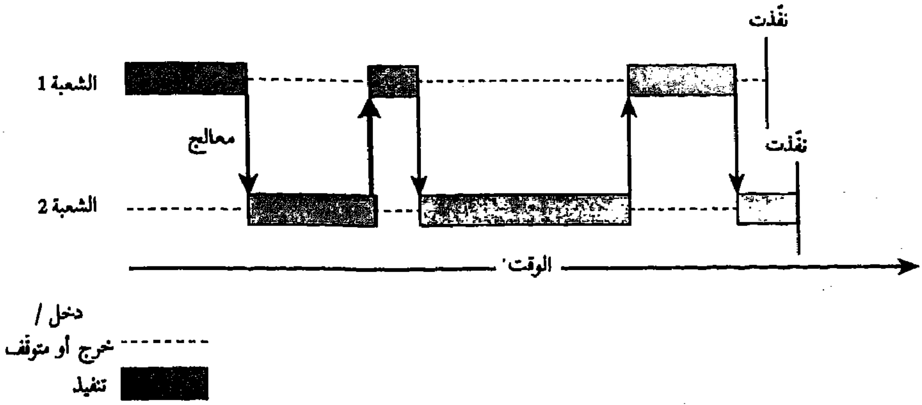
يستطيع المعالج تنفيذ فقط شعبة واحدة في كل مرة. لكن، يتيح نظام التشغيل المتعدّد المهام للمستعمل تشغيل عدّة برامج، حيث يبدو وكأنه ينفّذها كلها في نفس الوقت. وهو يحقّق ذلك بالطريقة التالية:

1 ـــ يشغّل شعبة إلى أن يقاطع تنفيذ الشعبة أو إلى أن تنتظر الشعبة توفّر مورد.
2 ـــ يحفظ سياق الشعبة.
3 ـــ يحمّل سياق شعبة آخر.
4 ـــ يكرّر هذا التتابع طالما يوجد شعب للتنفيذ.

يسمى تبديل تنفيذ المعالج من شعبة واحدة إلى أخرى بهذه الطريقة، تبديل السياق. وفي النظام Windows NT، تنفّذ عمليّة تبديل السياق بواسطة مكوّن النواة للبرنامج التنفيذي.

كما يوضح الشكل (4-4) على الصفحة التالية، مع وجود شعبتين، يبدّل باستمرار نظام تشغيل متعدّد المهام التنفيذ من شعبة واحدة إلى أخرى. وتنهي كل شعبة مهمتها الفرعيّة ثم إما تنتهي وإما تحدّد مهمّة أخرى لها. توفّر السرعة الفائقة للمعالج الإيحاء بأن كل الشعَب تنفّذ في نفس الوقت.

تزيد المهام المتعدّدة كميّة العمل الذي يقوم به النظام لأنه لا يمكن تنفيذ معظم الشعَب بشكل متواصل. تتوقف الشعبة دورياً عن التنفيذ وتنتظر أن ينهي جهاز دخل / خرج بطيء نقل بيانات أو خلال إستعمال شعبة أخرى لمورد تحتاجه. فعند ضرورة إنتظار شعبة واحدة، تتيح المهام المتعدّدة تنفيذ شعبة أخرى، مع إستعمال مزيّة دورات المعالج التي يمكن أن تُهدَر.



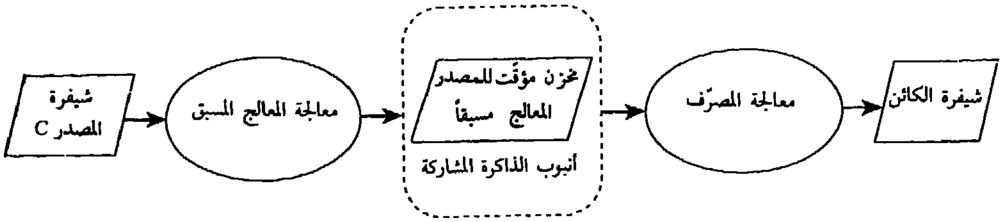
الشكل (4-4)
المهام المتعدّدة

المهام المتعدّدة الوقائيّة هي شكل من أشكال المهام المتعدّدة حيث لا ينتظر نظام التشغيل قيام شعبة بتوفير المعالج تلقائياً لشعَب أخرى. لكنه، يقاطع شعبة بعد إشتغالها لفترة زمنيّة مضبطة مسبقاً، تسمى كميّة الوقت، أو عندما تصبح شعبة بأولويّة أعلى (كتلك التي تستجيب لدخل المستعمل) جاهزة للتشغيل. يمنع حق الشفعة شعبة واحدة من إحتكار المعالج ويتيح للشعَب الأخرى المشاركة العادلة لوقت التنفيذ. إن البرنامج التنفيذي NT هو نظام متعدد المهام وقائي كمحيطه الأولي Windows أي النظام الفرعي Win 32. وفي إصدارات Windows غير الوقائيّة المعتمدة على النظام MS-DOS، تتخلّى شعبة تلقائياً عن التحكّم بالمعالج لكي تنفّذ المهام المتعدّدة.

تتطلّب شعبتان أحياناً القدرة على الإتصال مع بعضها البعض لتنسيق نشاطاتها باتجاه تحقيق الهدف المشترك. فمثلاً، قد يحتوي مصرّف C على شعبة واحدة تعالج مسبقاً برنامج C وشعبة أخرى تستلم خرج الشعبة الأولى وتصرّفه في شيفرة كائن. يجب أن تحتوي الشعبتين على طريقة لتمرير البيانات بينها.

حتى النصف الثاني من الثمانينات، أتاحت معظم أنظمة التشغيل للمعالجة الحصول على

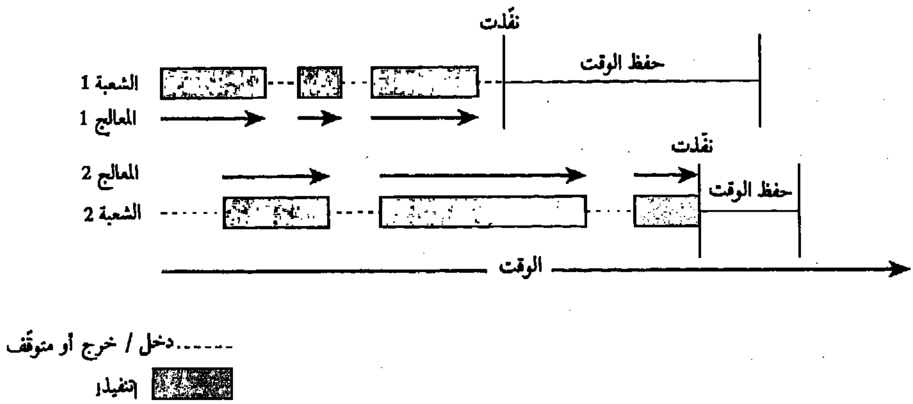
شعبة تنفيذ واحدة فقط. (وفي الواقع، إستعمل معظم أنظمة التشغيل التعبير معالجة للإشارة إلى وحدة مستقلّة قابلة للتنفيذ. والشعبة هو تعبير جديد).



الشكل (4-5)
مصرّف معالجتين

ولأن كل معالجة كانت تحتوي على فسحة عنوان مستقلّة، وجب على المعالجتين تحقيق منطقة ذاكرة مشاركة أو ملفّاً مشاركاً إذا أرادتا الإتصال مع بعضها البعض. وقد استعملت الأنابيب عموماً لتحقيق هذا النوع من الإتصال داخل المعالجة. راجع الشكل (4-5).

إن إستعمال معالجتين (كلٌّ منها بشعبة واحدة) لمعالجة برنامج مسبقاً. وتصريفه هو أسرع من إستعمال معالجة واحدة لأن نظام التشغيل المتعدّد المهام يورّق بينياً تنفيذ شعبة المعالج المسبق وشعبة المصرّف. وحالما يضع المعالج المسبق أي شيء في المخزن المؤقّت المشارك، يبدأ المصرّف عمله. تسمّى التطبيقات المنفّذة في موقعين أو أكثر، التطبيقات المتزامنة.



الشكل (4-6)
المعالجة المتعدّدة

التزامن في تطبيق هو أمر مفيد على الحاسوب الأحادي المعالج لكنه أكثر إفادة على الحاسوب المتعدّد المعالجة. وبوجود عدّة معالجات، يمكن أن يعمل المعالج المسبق والمصرّف في تزامن. فإذا صمّم تطبيق متزامن وخفّض سعي شعبة إلى الموارد، فإنه يمكن أن ينفذ بسرعة أكبر على حاسوب متعدّد المعالجات المقارنة على حاسوب أحادي المعالج. يوضّح الشكل (6-4) عند مقارنته مع الشكل (4-4)، هذه النقطة.

إن نظام التشغيل المتعدّد المعالجة مصمّم بشكل خاص ليشتغل على حواسيب بأكثر من معالج واحد. يستطيع نظام تشغيل متعدّد المعالجة متناظرة (SMP)، مثل النظام Windows NT، تشغيل شيفرة نظام التشغيل وشيفرة المستعمل على أي معالج متوفّر. وعند توفُّر عدّة شعَب للتشغيل أكثر من توفُّر المعالجات، ينفّذ نظام تشغيل SMP أيضاً المهام المتعدّدة حيث يقسّم كل وقت معالج على كل الشعَب المنتظرة. (راجع الفصل السابع، «النواة» لمزيد من المعلومات حول جدولة الشعَب على النظام Windows NT).

2-2-4 الشعَب المتعدّدة:

لا يكفي دائماً إستعمال معالجتين لتحقيق التزامن. فعلى بعض أنظمة UNIX، مثلاً، وعندما تنشىء معالجة واحدة معالجة أخرى، يجب على النظام نسخ كل شيء موجود في فسحة عنوان المعالجة الأولى إلى فسحة عنوان المعالجة الجديدة. وهذه العمليّة تستغرق وقتاً كبيراً، خاصة لفسحة العنوان الكبيرة. إضافة لذلك يجب أن تحقّق المعالجتين طريقة لمشاركة البيانات وهي وظيفة سريعة وسهلة على بعض أنظمة التشغيل لكنها ليست كذلك على بعض الأنظمة الأخرى. يعالج النظام Windows NT هذه المشاكل عن طريق توفير آليّة مناسبة لمشاركة الذاكرة، باستعمال ذاكرة النسخ عند التعديل لتجنُّب نسخ فسحة عنوان بأكملها من معالجة واحدة إلى أخرى وعن طريق تطبيق برنامج خدماتي لتمرير الرسائل مستمثل محليًّا. (يتمّ شرح القُدرتين الأوّليتين في الفصل السادس «برنامج إدارة الذاكرة الظاهريّة» ويتمّ شرح القدرة الثالثة ــ البرنامج الخدماتي لإستدعاء إجراء محلّي (LPC) ــ في الفصل الخامس «النظام Windows والأنظمة الفرعيّة المحميّة»).

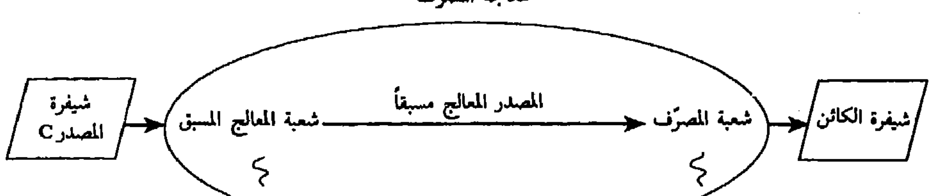
وحتى مع هذه التحسينات، توجد أوقات يكون فيها إستعمال طريقة أخرى مفيداً ــ بالتحديد، المعالجات المتعدّدة الشعَب. وكما ذكر سابقاً، يشير التعبير شعبة إلى تحرُّك معالج عبر تعليمات برنامج. فكل شعبة تمثّل عدّاد برنامج مستقلّ. تحتوي المعالجة المتعدّدة الشعَب على شعبتين أو أكثر (وعدّادات البرنامج) ضمن معالجة واحدة، حيث تشارك نفس فسحة العنوان ومقابض الكائن والموارد الأخرى.

تنشأ كل معالجة NT بواسطة شعبة واحدة. ويستطيع البرنامج إنشاء شعَب إضافيّة في المعالجة حسب الحاجة. تستعمل هذه الشعَب الإضافيّة غالباً للعمليات غير المتزامنة في برنامج ـ أي، العمليات التي تحصل في أي وقت دون إعتبار لأسباب البرنامج الرئيسي. وتتلاءم عمليات الدخل / الخرج في غالب الأحيان في هذه الفئة. فمثلاً، قد تستعمل شعبة لتحفظ دورياً مستنداً قيد التحرير أو لمراقبة جهاز، مثل لوحة المفاتيح أو ماوس، لإدخال المستعمل. وبإستعمال شعبة واحدة لتشغيل البرنامج الرئيسي وإنشاء شعبة أخرى لمراقبة جهاز، يستطيع النظام جدولة العمليتين بشكل مستقلّ عن معالج، حيث تنفّذ المهام المتعدّدة. عند التشغيل على حاسوب متعدّد المعالجات، يمكن تنفيذ الشعبتين في نفس الوقت دون الحاجة لإنشاء معالجة ثانية وتحفيز فسحة عنوانها.

لتحقيق التزامن بإستعمال الشعَب، ينشيء برنامج شعبتين أو أكثر لتنفيذ الأقسام المختلفة من برنامجها ضمن نفس المعالجة. يرسم مصرّف متعدّد الشعَب في الشكل (4-7).

تحقّق المعالجات المتعدّدة الشعَب، التزامن دون الحاجة لإستعمال معالجتين. فالشعَب أقلّ علواً وهي تنشيء بسرعة أكبر من إنشاء المعالجات. (وهي تسمى أحياناً «معالجات خفيفة الوزن» لهذا السبب) ولأن كل الشعَب في معالجة تشارك نفس الذاكرة باستثناء تكديسها ومحتويات المسجل، لا حاجة لآلية تمرير بيانات خاصة. فشعبة واحدة تكتب خرجها إلى الذاكرة وشعبة أخرى تقرأه كدخل. وبشكل مشابه، تتوفّر بالتساوي كل موارد المعالجة (الكائنات) لكل الشعَب في المعالجة.

تستعمل النواة NT مخطّط أولويّة لإنتقاء ترتيب تنفيذ الشعَب. تنفّذ الشعَب بأولويّة أعلى قبل الشعَب بأولويّة أدنى وتغيّر النواة أولويّة الشعبة دورياً لضمان تنفيذ كل الشعَب. ويتيح برنامج تطبيقي تنفيذ شعبة على أي معالج في حاسوب متعدّد المعالجات أو يستطيع حصر تنفيذها على مجموعة فرعيّة من المعالجات.



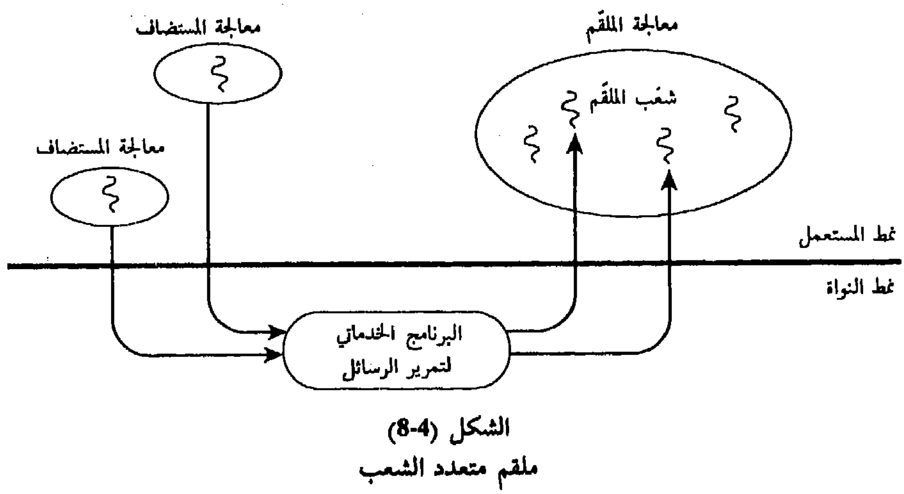
الشكل (4-7)
مصرّف متعدّد الشعَب

إن إنشاء معالجة متعدّدة الشعَب هو حلٌ مثالي لتطبيقات الملقّم (مثل الأنظمة الفرعيّة المحميّة للنظام Windows NT) الذي يقبل الطلبات من المستضافات وينفّذ نفس الشيفرة لكل طلب. فمثلاً، ينفّذ ملقّم ملفّ العمليات على الملفّات وهو يفتح الملفّات ويقرأ منها ويكتب إليها ويغلقها. ورغم أن كل معالجة قد تتطلّب من الملقّم العمل على ملفّ مختلف، يحمّل البرنامج الملقّم في الذاكرة لمرّة واحدة فقط. يستلم كل طلب داخل ويناول من قبَل شعبة ملقّم مستقلّ الذي ينفّذ وظيفة الملقّم المناسبة. يوضّح الشكل (4-8) على الصفحة التالية هذه النقطة.

في هذا الشكل، تستعمل معالجتا مستضاف (كل معالجة بشعبة واحدة ممثّلة بخط خربشة)، البرنامج الخدماتي لتمرير الرسائل وذلك لإرسال رسالة إلى معالجة الملقّم. تتوفّر عدّة شعَب ملقّم لتنفيذ شيفرة الملقّم والإستجابة للمستضافات.

لاحظ أن كتابة تطبيقات متعدّدة الشعَب تتطلّب إنتباهاً كبيراً لأن كل الشعَب ضمن معالجة تحتوي وصولاً كاملاً إلى فسحة عنوان المعالجة. وقد تتعثّر الشعَب عرضياً وتصطدم ببعضها البعض، حيث تقرأ الذاكرة أو تكتب إليها خارج دورها.

وهذه ليست حالة التطبيقات التي تستعمل معالجتين لتحقيق التزامن والتي تتّصل عبر الرسائل أو الأنابيب. ولا يمكن لمعالجة واحدة إتلاف فسحة عنوان معالجة أخرى أو تشويهها عن قصد أو عرضياً. لهذا السبب تستخدم الأنظمة الفرعيّة المحميّة Windows NT كمعالجات ملقّم مستقلّة (ولهذا تسمّى أنظمة فرعيّة «محميّة»). ويحافظ كل نظام فرعي على التحكُّم بفسحة العنوان الخاصة به، دون تدخُّل من قبَل الأنظمة الفرعيّة الأخرى أو من قبَل المعالجات. ولكن ضمن الملقّم، يفضّل تشغيل الشعَب المتعدّدة. ومشاركة نفس فسحة العنوان والموارد.

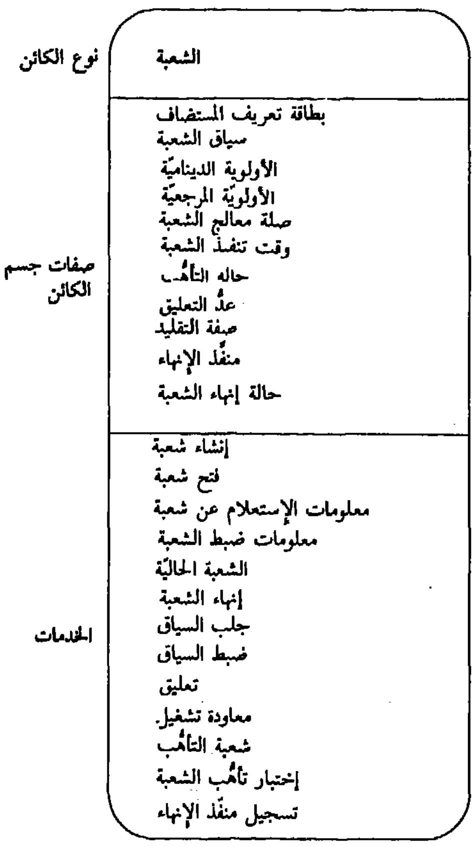


الشكل (4-8)
ملقم متعدد الشعب

إن تحقيق التزامن باستعمال معالجات متعدّدة وبإستعمال شعَب متعدّدة ضمن معالجة هي طرق مفيدة. وتحدّد أهداف البرنامج التطبيقي البنية الأكثر إفادة في أي برنامج معينّ.

بإيجاز، تشير البنود التالية إلى تطبيق المعالجات من قبَل نظام تشغيل:

- المهام المتعدّدة. تقسيم وقت المعالج ضمن الشعَب التي تنتظر التنفيذ وإنشاء وضمّ تنفيذ كل الشعَب في نفس الوقت.
- المعالجة المتعدّدة. تشغيل نفس شيفرة نظام التشغيل على حواسيب أحاديّة المعالج ومتعدّدة المعالج. ويشغل نظام تشغيل متعدّد المعالجة متناظر شيفرة النظام وشيفرة المستعمل على كل المعالجات المتوفّرة.



الشكل (4-9)

كائن الشعبة

■ الشعَب المتعدّدة. دعم شعبة واحدة أو أكثر ضمن معالجة واحدة.

يجب على نظام التشغيل المتقدّم تزويد كل هذه القدرات والنظام Windows NT يوفّرها.

3-2-4 كائن الشعبة:

تبقى المعالجة NT غير نافعة إلى أن تجدول شعبة للتنفيذ. وحالما تحتوي معالجة على شعبة، تستطيع تلك الشعبة إنشاء شعَب إضافيّة.

ومثل المعالجات، تستخدم شعَب البرنامج التنفيذي NT ككائنات، تم إنشاءها وحذفها من قبَل برنامج إدارة الكائنات. يعرّف برنامج إدارة المعالجة جسم كائنات الشعبة وخدمات النظام المستعملة لمناولة الشعَب بعد إنشائها. يرسم كائن الشعبة في الشكل (4-9).

كما يُشاهَد، تشبه بعض الصفات في كائن الشعبة تلك في كائن المعالجة. بعض الصفات

الصفة	الغرض
بطاقة تعريف المستضاف	قيمة فريدة تعرّف شعبة عندما تستدعي ملقّماً.
سياق الشعبة	مجموعة قيم مسجّل وبيانات متطايرة أخرى تعرف حالة تنفيذ شعبة.
الأولويّة الديناميّة	أولويّة تنفيذ الشعبة في أية لحظة محدّدة.
الأولويّة المرجعيّة	الحدُّ السفلي للأولويّة الديناميّة للشعبة.
صلة معالج شعبة	مجموعة المعالجات حيث يمكن أن تشتغل الشعبة مجموعة فرعيّة (غير صحيحة) من صلة المعالج لمعالجة شعبة.
وقت تنفيذ الشعبة	كميّة الوقت المتراكمة التي إستغرقتها شعبة في نمط المستعمل وفي نمط النواة.
حالة التأهُّب	عِلمٌ يشير إلى ضرورة تنفيذ شعبة لإستدعاء إجراء غير متزامن (APC).
عدُّ التعليق	عدد مرّات تعليق تنفيذ الشعبة دون معاودتها.
صفة التقليد	صفة وصول مؤقتة تتيح لشعبة تنفيذ عمليات معالجة أخرى (مستعملة من قبَل الأنظمة الفرعيّة).
منفّذ الإنهاء	قناة إتصال داخل العمليّة يرسل إليها برنامج إدارة المعالجة رسالة عند إنتهاء الشعبة (مستعملة من قبَل الأنظمة الفرعيّة).
حالة خروج الشعبة	سبب إنهاء الشعبة.

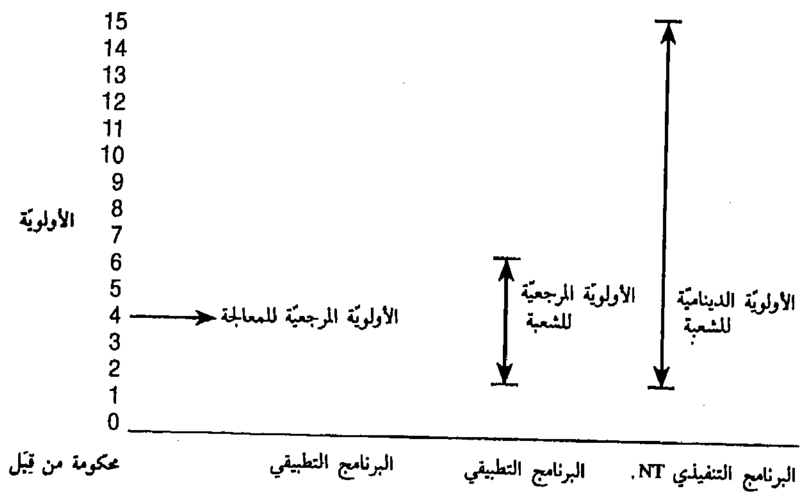
الجدول (4-2)
صفات كائن الشعبة

المعنيّة، مثل صلة معالج الشعبة والأولويّة الديناميّة، تقيّد فعليًّا أو تؤهّل القيم المطبّقة على المعالجة ككلّ. فمثلاً، تتّصف كل شعبة بصلَة معالج والتي هي مجموعة فرعيّة غير صحيحة (تساوي أو أقلّ من) صلة المعالج المعيّنة لمعالجته. لذلك، يمكن تشغيل بعض الشعَب المختلفة ضمن معالجة قسرياً على مجموعات فرعيّة مختلفة من المعالجات.

بشكل مشابه، تحتوي كل شعبة على أولويّة تنفيذ مرجعيّة تمتدُّ من مستويين تحت الأولويّة المرجعيّة للمعالجة إلى مستويين فوقها، كما يبينّ في الشكل (4-10).

وكما يبينّ في الشكل، تحتوي كل شعبة أيضاً على أولويّة ديناميّة تبدأ عند الأولويّة المرجعيّة للشعبة وتختلف صعوداً وفقاً لنوع العمل المنفّذ من قبَل الشعبة. فمثلاً، إذا إستجابت شعبة لدخل مستعمل، ترفع النواة NT أولويّتها الديناميّة. وإذا كانت مربوطة حسابياً تخفّض النواة تدريجياً أولويّتها الديناميّة إلى أولويّتها المرجعيّة. وبتخفيض الأولويّة المرجعيّة لشعبة واحدة ورفع الأولويّة المرجعيّة لشعبة أخرى، تستطيع الأنظمة الفرعيّة التحكُّم بالأولويّات المتعلّقة للشعب ضمن معالجة. تتحكُّم الأولويّة المرجعيّة للمعالجة بتباعد أولويّات الشعَب ضمن المعالجة وبكيفيّة تعلُّق أولويّات الشعَب مع أولويّات المعالجات الأخرى.

ومثل أولويّات الشعبة، تتواجد صفات أخرى في كائن الشعبة لتتيح لنظام التشغيل (وخاصة الأنظمة الفرعيّة للمحيط) التحكُّم بالشعَب التي تنشئها. فمثلاً، تحتوي صفة سياق الشعبة كل ما يحتاج لمعرفته نظام التشغيل لمتابعة تنفيذ شعبة بعد مقاطعتها ـ وبالتحديد، القيم



الشكل (4-10)
علاقات الأولويّة

المخزّنة في مسجلات المعالج وعلى تكديسات نمط المستعمل ونمط النواة للشعبة. وبتعليق شعبة، وتعديل سياق نمط المستعمل، تمَّ إعادة بدء الشعبة، يستطيع نظام فرعي لمحيط تعديل تصرّف الشعبة أو بدء تنفيذها عند موقع يختلف عن مكان تعليقها. (تستطيع مزيلات العلل من نمط المستعمل إستعمال هذه القدرة للحكم بتنفيذ الشعَب).

إن تنبيه شعبة، خدمة أخرى متوفّرة لكائنات الشعبة، هي قدرة تتيح لنظام فرعي لمحيط أو لأقسام أخرى من نظام التشغيل، إبلاغ بعض الشعَب بشكل غير متزامن لضرورة تنفيذ إجراء خاص. وتستطيع الشعبة التي يفترض أن تكون متأهّبة، إستدعاء خدمة لإختبار تعليق التأهُّب. (راجع القسم 4-2-5).

يشبه منفّذ إنهاء الشعبة منافذ الإستثناء وإزالة العلل لمعالجة. يتيح منفّذ الإنهاء إبلاغ نظام فرعي لمحيط عند إنهاء شعبة في إحدى معالجات المستضاف. وهو يستطيع بعد ذلك تحديث أي معلومات إحتفظ بها حول الشعبة أو المعالجة حيث تستقرُّ الشعبة.

تتيح خدمة الشعب الحاليّة حصول شعبة على مقبض بسرعة دون فتح واحدة. وهي تستطيع إستعمال المقبض. مثلاً، لإسترداد المعلومات حول نفسها مثل وقت التنفيذ الإجمالي وأولويّة التنفيذ الحاليّة وصلة المعالج.

توفّر الأقسام اللاحقة معلومات إضافية حول خدمات المعالجة والشعبة. ويصف الفصل التالي الأنظمة الفرعيّة المحميّة Windows NT.

4-2-4 المزامنة:

عند إشتغال تطبيق متزامن، تحتاج شعبة في غالب الأحيان لطريقة إتصال مع بعضها لتنسيق نشاطاتها. إن تمرير الرسائل عبر الأنابيب هو أحد الأمثلة عن الإتصال. لكن أبسط شكل إتصال هو المزامنة. وتنسب المزامنة إلى قدرة شعبة واحدة على توقيف التنفيذ إدارياً، وإنتظار تنفيذ شعبة أخرى لبعض العمليات.

في أمثلة المصرّف التي عُرضت سابقاً، يقرأ المعالج المسبق شيفرة المصدر C ويكتب خرجه إلى مخزن ذاكرة مؤقت بحيث يتشارك مع المصرّف. يقرأ المصرّف هذا الخرج كدخله ويصرّفه وينشىء شيفرة كائن. وعندما يبدأ تشغيل البرنامج، يجب أن تنتظر شعبة المصرّف إلى أن تضع شعبة المعالج المسبق شيئاً ما في المخزن المؤقّت قبل محاولتها قراءته. وبشكل مشابه، إذا امتلأ المخزن المؤقّت، يجب أن ينتظر المعالج المسبق إلى أن يزيل المصرّف البيانات من المخزن المؤقت قبل وضع مزيد من البيانات فيه.

يجب أن توفّر كل أنظمة التشغيل المتعدّدة المهام أو المتعدّدة المعالجة طريقة لكي تنتظر الشعَب شعبة أخرى للقيام بشيء. فمثلاً، لإفلات سواقة شريطيّة أو لإنهاء كتابة مخزن ذاكرة مشارك. كذلك، يجب أن يتيح نظام التشغيل لشعبة لتشير إلى شعَب أخرى أنهت مثل هذه العمليّة. وبعد إبلاغها، يمكن أن تتابع شعبة منتظرة التنفيذ.

في البرنامج التنفيذي NT، تستخدم قدرات الإنتظار والتأشير هذه، كجزء من تصميم الكائن. فكائنات المزامنة هي كائنات تنفيذيّة تزامن معها شعبة تنفيذها. تشمل كائنات المزامنة ما يلي:

- كائنات المعالجة
- كائنات الشعبة
- كائنات الملفّ
- كائنات الحدث
- كائنات زوج الأحداث
- كائنات الإعلام الإشاري
- كائنات المؤقّت
- كائنات الخافت

تخدم أول ثلاث كائنات مسردة أغراضاً أخرى إضافة إلى المزامنة، لكن آخر خمس كائنات تتواجد لدعم المزامنة. سويّة، تتيح هذه الكائنات التنفيذيّة الشعَب تنسيق نشاطاتها مع مجموعة من تزامنات النظام، حيث تطبق قواعد مختلفة على حالات مختلفة.

وفي أية لحظة معيّنة، يكون كائن مزامنة في حالة من حالتين، أما في حالة مشار إليها، وفي حالة غير مشار إليها. تعرّف الحالة المشار إليها بشكل مختلف للكائنات المختلفة. يكون كائن شعبة في حالة غير مشار إليها خلال مدّة خدمته ويضبط إلى الحالة المشار إليها بواسطة النواة NT عندما تنتهي الشعبة. وبشكل مشابه، تضبط النواة كائن معالجة إلى الحالة المشار إليها عندما تنتهي آخر شعبة للمعالجة. وبالعكس، يضبط كائن المؤقّت، كساعة التوقيت، لينطلق في وقت معيّن. وعندما ينفّذ وقته، تضبط النواة كائن الموقّت إلى الحالة المشار إليها.

للمزامنة مع كائن، تستدعي الشعبة إحدى خدمات نظام الإنتظار المزوّدة من قبَل برنامج إدارة الكائنات، حيث تمرّر مقبضاً إلى الكائن المطلوب المزامنة معه. وتستطيع الشعبة إنتظار كائن واحد أو عدّة كائنات ويمكنها تحديد إلغاء إنتظارها إذا لم تنتهي في خلال فترة زمنيّة معيّنة.

وعندما تضبط النواة كائناً إلى الحالة المشار إليها، فإنها تدقّق لجهة إنتظار أي شعَب للكائن. فإذا كانت الحالة كذلك، تفلت النواة شعبة واحدة أو أكثر من حالة الإنتظار لكي تتابع التنفيذ.

عند إختيار آليّة مزامنة، يجب أن يعتبر البرنامج قواعد تنظيم تصرُّف كائنات المزامنة المختلفة. ويغيّر إنتهاء إنتظار شعبة عند ضبط كائن إلى الحالة المشار إليها مع نوع الكائن الذي تنتظره الشعبة، كما يوضّح ذلك الجدول (3-4) على الصفحة التالية.

عند ضبط كائن إلى حالة مشار إليها، تفلت الشعَب المنتظرة عادة في حالة الإنتظار فوراً. فمثلاً، يستعمل كائن حدث لإعلان حصول حدث ما. وعند ضبط كائن الحدث إلى الحالة المشار إليها، تفلت كل الشعَب المنتظرة على الحدث. أما الإستثناء فهو أية شعبة تنتظر أكثر من كائن واحد في كل مرة. وقد تواصل هذه الشعبة إنتظارها إلى أن تبلغ الكائنات الإضافيّة الحالة المشار إليها.

وبعكس كائن حدث، يحتوي كائن خافت (مرئي ككائن خافت لمبرمجي Win 32) على ملكيّة متعلّقة به. وهو يستعمل للوصول الحصري التبادلي إلى مورد، وتستطيع شعبة واحدة فقط في كل مرة الحصول على كائن خافت. وعندما يصبح الكائن الخافت حراً، تضبطه النواة إلى الحالة المشار إليها ثم تنتقي شعبة واحدة منتظرة لتنفيذها. تكتب الشعبة المنتقاة بواسطة النواة، الكائن الخافت، وتواصل كل الشعَب المتبقّية الإنتظار (يصف الفصل السابع «النواة» المزامنة في تفصيل أكبر).

نوع الكائن	يضبط إلى الحالة المشار إليها عندما	التأثير على الشعَب المنتظرة
معالجة	تنتهي الشعبة الأخيرة	تفلت كلها
شعبة	تنتهي الشعبة	تفلت كلها
ملفّ	إكمال عمليّة الدخل / الخرج	تفلت كلها
حدث	تضبط الشعبة الحدث	تفلت كلها
زوج أحداث	تضبط شعبة مستضاف أو ملقّم مكرّسة الحدث	تفلت الشعَب المكرّسة الأخرى
إعلام إشاري	يهبط عدُّ الإعلام الإشاري إلى الصفر	تفلت كلها
مؤقّت	بلوغ الوقت المضبط أو نفاذ الفترة الزمنيّة	تفلت كلها
خافت	تفلت الشعبة الخافت	تفلت شعبة واحدة

الجدول (3-4)
تعريفات الحالة المشار إليها

إن عِلم الألسنيّة المزامنة للبرنامج التنفيذي NT واضحة لمبرمجي Win 32 عبر روتينات API () WaitForSingleObject و () WaitForMultipleObjects، التي يستخدمها النظام الفرعي Win 32 باستدعاء خدمات نظام تماثليّة مزوّدة من قبل برنامج إدارة الكائنات NT. تستطيع شعبة في برنامج تطبيقي Win 32 التزامن مع كائن معالجة Win 32 وشعبة وحدث وإعلام إشاري وخافت أو ملفّ. فمثلاً، قد تتزامن شعبة مع شعبة أخرى في برنامج صفحة جدوليّة. إفترض أن البرنامج التطبيقي يحتوي على شعبة رئيسيّة تنفّذ وظائف صفحة جدوليّة عاديّة وشعبة ثانويّة ترصف ملفّات الصفحة الجدوليّة إلى الطابعة. وافترض الآن أن المستعمل يطبع صفحة جدوليّة وقبل إتمام الرصف، أدخل أمراً لإنهاء البرنامج. لكن الشعبة الرئيسيّة التي تقبل طلب الإنهاء، لن تنهي المعالجة فوراً (رغم أنها قد تخلي الشاشة). لكن وبدلاً من ذلك، فإنها تستدعي الروتين () WaitForSingleObject لينتظر أن تنتهي شعبة الراصف من الرصف والإنهاء. وبعد الإنهاء، تفلت الشعبة الرئيسيّة من عمليّة الإنتظار وتنهي نفسها الأمر الذي ينهي برنامج الصفحة الجدوليّة وينهي معالجة الصفحة الجدوليّة.

5-2-4 التنبيهات وإستدعاءات الإجراء اللامتزامن:

في بعض الحالات، من المفيد الإتاحة لشعبة واحدة إبلاغ شعبة أُخرى لا تزامنياً لإيقاف ما تقوم به. تتعلّق هذه العمليّة، التي تسمى تأهُب في البرنامج التنفيذي NT، بالمزامنة. إفترض أن برنامج قاعدة بيانات يستجيب لعملية إستعلام. وهو لا يعرف إذا كانت البيانات المطلوبة متوفّرة على حاسوب محلّي أو على حاسوب بعيد. ولمعرفة ذلك، فإنه يبدأ تشغيل شعبتين، واحدة تبحث عن البيانات محلياً والأخرى تبحث عن البيانات على الشبكة. وحالما تجد شعبة واحدة البيانات، فإنها تنبّه الشعبة الأخرى. واستجابةً لذلك، تتوقّف الشعبة التي تلقّت التنبيه عن ما تقوم به وترجع لتستعدّ للقيام بمهمّة جديدة.

لا تستعمل قدرة التنبيه بكثرة في النظام Windows NT إلا بالدمج مع آلية إبلاغ لا متزامنة أخرى، تسمى إستدعاء الإجراء اللامتزامن (APC). ومن وقت لآخر، يحتاج نظام التشغيل لإبلاغ شعبة إلى ضرورة تنفيذ مهمّة معيّنة. وفي بعض الأحيان، تعمل الشعبة بعد حصول حدث. فمثلاً، يستطيع مستعمل الطلب من Windows إرسال رسالة تذكّره بأوقات المواعيد المجدولة. وفي النظام Windows NT، يتمّ هذا النوع من الإبلاغ بإستخدام إجراء APC في نمط المستعمل ــ أي، يستدعي النظام الفرعي Win 32 البرنامج التنفيذي NT لضبط مؤقّت وتوفير مؤشّر إلى إجراء (APC) يرسل رسالة إلى المستعمل. وعندما ينطلق المؤقّت، يحثّ البرنامج التنفيذي NT شعبة النظام الفرعي Win 32. بعد ذلك، تتابع شعبة Win 32 ما كانت تفعله.

رغم أن بعض العمليات اللامتزامنة تولّد بواسطة برامج في نمط المستعمل، حيث يولّد معظمها بواسطة نظام التشغيل، وخاصة بواسطة نظام الدخل / الخرج NT. إن نظام الدخل / الخرج NT لا متزامن وهذا يعني أن المستدعي يستطيع بدء عملية دخل / خرج ثم القيام بأعمال أخرى خلال إتمام الجهاز العملية. وعندما ينتهي الجهاز من نقل البيانات، يجب أن يقاطع نظام الدخل / الخرج ماتفعله الشعبة المستدعية وينسخ نتائج عملية الدخل / الخرج إلى فسحة عنوان الشعبة. يستعمل نظام الدخل / الخرج إجراء APC في نمط النواة لتنفيذ هذه العمليّة.

تختلف إجراءات APC في نمط المستعمل وفي نمط النواة في عدّة نواحي، لكن يوجد إختلاف واحد واضح؛ فإجراء APC في نمط النواة يستطيع مقاطعة تنفيذ شعبة في نمط المستعمل في أي وقت وجعلها تنفّذ الإجراء. وهذا يحصل عادةً بطريقة مجهولة لبرنامج تطبيقي. يحصل مقاطعة لبرامجيات، كما في مقاطعة العتاد، و«يسرق» النظام شعبة البرنامج التطبيقي لفترة قصيرة ويجعلها تنفّذ إجراء APC. لكن وبالمقابل، فإن إجراء APC في نمط المستعمل يسلّم فقط عند نقاط التحكُّم عندما تكون الشعبة التي طلبته جاهزة لتنفيذه.

يوفّر NT طريقتين تستطيع شعبه بواسطتها التحكُّم عند إستلامها إبلاغ لا متزامن في نمط المستعمل (تنبيه أو إجراء APC في نمط المستعمل). وتستطيع الشعبة إما إستدعاء خدمة محليّة للتأكُّد من تنبيهها، أو يمكنها إنتظار مقبض كائن، حيث تحدّد أنه يمكن مقاطعة إنتظارها بواسطة تنبيه. في أية حالة، إذا كان إجراء APC في نمط المستعمل في إنتظار الشعبة، فإن النواة NT تسلّمه وتنفّذ الشعبة الإجراء. بعد ذلك، تعاود النواة تنفيذ الشعبة من نقطة مقاطعتها.

تجعل روتينات Win 32 API التنبيهات وإجراءات APC مرئيّة عبر روتينات الدخل / الخرج الممدّدة (NT فقط). وتتيح روتينات ReadFileEx () API و WriteFileEx () لشعبة قراءة ملفّ أو الكتابة إليه لا تزامنياً، حيث تزوّد روتين APC تنفّذه الشعبة بعد إتمام عملية الدخل / الخرج. وتتيح روتينات WaitForSingleObjectEx () و WaitForMultipleObjectsEx () لشعبة الإنتظار في حالة تأهُّب عند بعض النقاط بعد إصدار إستدعاء دخل / خرج. لا يوفّر النظام الفرعي POSIX قدرات APC إلى تطبيقات POSIX لكنه يستعمل إجراءات APC في نمط النواة لمحاكاة تسليم إشارة POSIX إلى معالجات POSIX. وبشكل مشابه، تستطيع الأنظمة الفرعيّة المستقبليّة إستعمال إجراءات APC لإستخدام برامج خدماتيّة أخرى للإبلاغ اللامتزامن. يعاود موضوع إجراءات APC الظهور في شروح لاحقة للنواة NT، والتي تتحكّم بمعالجة APC ونظام الدخل / الخرج NT الذي يستعمل إجراءات APC بشكلٍ مكثّف.

3-4 بنية المعالجة:

المعالجات هي وحدات مستقلّة ديناميّة، تنشأ وتُتلف خلال إشتغال نظام التشغيل. تنشىء معالجة واحدة معالجة أخرى والتي بدورها تنشىء معالجات أخرى. ينسب التعبير بنية المعالجة إلى كيفيّة إنشاء نظام التشغيل المعالجات والشعَب وإدارتها والتخلُّص منها إلى كيفيّة تعلُّق معالجة واحدة مع المعالجات الأخرى خلال وجودها.

لا يشاهد المبرمجون الذين يكتبون برامج تطبيقيّة Win 32 و MS-DOS و OS/2 أو POSIX المعالجات المحليّة للبرنامج التنفيذي NT وشعبه. فالنظام الفرعي Win 32 والأنظمة الفرعيّة الأخرى تحجبها عن المبرمجين حيث تنشىء محيطات خاصة يشاهد فيها مبرمج Win 32 فقط معالجات مشابهة للنظام Win 32 ويشاهد مبرمج POSIX فقط معالجات مشابهة للنظام POSIX وهكذا. لكن القدرات الأساسيّة لبنية معالجة البرنامج التنفيذي NT هي التي تتيح لهذه المحيطات بالتواجد في نفس نظام التشغيل.

يشرح القسم التالي بعض متطلّبات الأنظمة الفرعيّة للمحيط المتنوّعة ويصف القسم اللاحق الآليات التي يوفّرها برنامج إدارة المعالجة NT للأنظمة الفرعيّة.

1-3-4 متطلّبات النظام الفرعي للمحيط:

إن إحدى المهام الرئيسيّة لنظام فرعي لمحيط Windows NT هي محاكاة روتينات API التي تتوقّعها تطبيقات المستضاف للنظام الفرعي (مثلاً، روتينات Win 32 أو POSIX API). الوظيفة الرئيسيّة الأخرى هي في إستخدام بنيات المعالجة المطلوبة من قبَل المستضافات. لقد رأى كل من Mark Lucovsky و Steve Wood، الذين صمّما النظام الفرعي لمحيط POSIX و OS/2 الأصلي في Windows NT، ما قد تطلبه قدرات الأنظمة الفرعيّة الحاليّة والمستقبليّة وذلك لمحاكاة روتينات API المتعلّقة. وقاما بتعريف القدرات المتعلّقة بالمعالجة التالية المطلوبة لمحيط نموذجي:

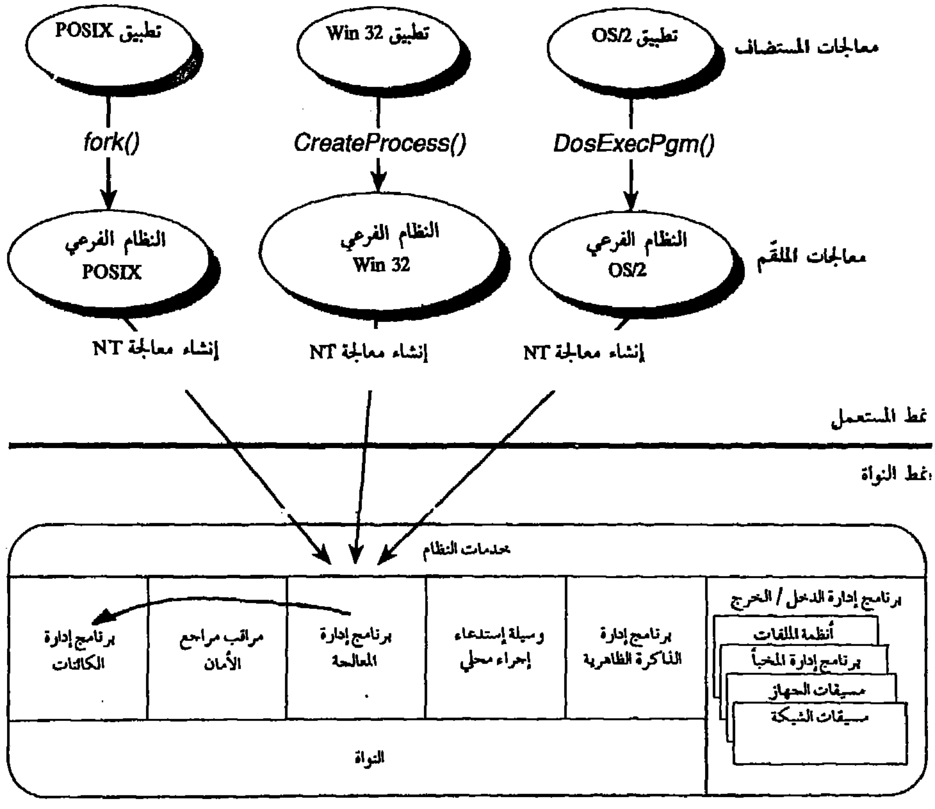
- إنشاء المعالجات والشعَب وإنهائها.
- تسجيل العلاقات والمحافظة عليها بين المعالجات.
- تنفيذ العمليات (المحليّة وعلى الشبكة) بدلاً من معالجة مستضاف.
- القراءة إلى فسحة عنوان معالجة والكتابة منها ومناولتها.
- توقيف شعبة مستضاف عن طريق تنبيه سياقها في نمط المستعمل وإعادة بدء تشغيلها.
- إلتقاط الإستثناءات الصادرة عن معالجات المستضاف ومناولتها.

إن إنشاء المعالجة، البند الأول في القائمة، هي عمليّة عامة لنظام فرعي توضّح طريقة

تنفيذ الأنظمة الفرعيّة لمحيط عملها بإستعمال خدمات المعالجة المحليّة. يرسم الشكل (4-11)، المبيّن على الصفحة التالية، العلاقة بين إنشاء معالجة من برنامج تطبيقي وإنشاء معالجة برنامج تنفيذي NT.

ينشىء تطبيق مستضاف ـ Win 32 أو OS/2 في هذا المثال ـ معالجة باستعمال روتين API مناسب لمحيطها، يرسل إستدعاء إنشاء المعالجة بواسطة البرنامج الخدماتي لتمرير الرسائل للبرنامج التنفيذي NT (الذي يشرح في الفصل الخامس «النظام Windows والأنظمة الفرعيّة المحميّة»)، إلى الملقّم المناسب الذي يستدعي برنامج إدارة المعالجة NT لإنشاء معالجة محليّة.

بعد إنشاء معالجة محليّة، يرجع برنامج إدارة المعالجة NT مقبضاً إلى كائن معالجة. يأخذ النظام الفرعي للمحيط المقبض وينشىء قيمة الرجيع المناسبة المتوقّعة من قبَل تطبيق المستضاف الأصلي. يبيّن ما يرجعه كل نظام فرعي في الشكل (4-12).

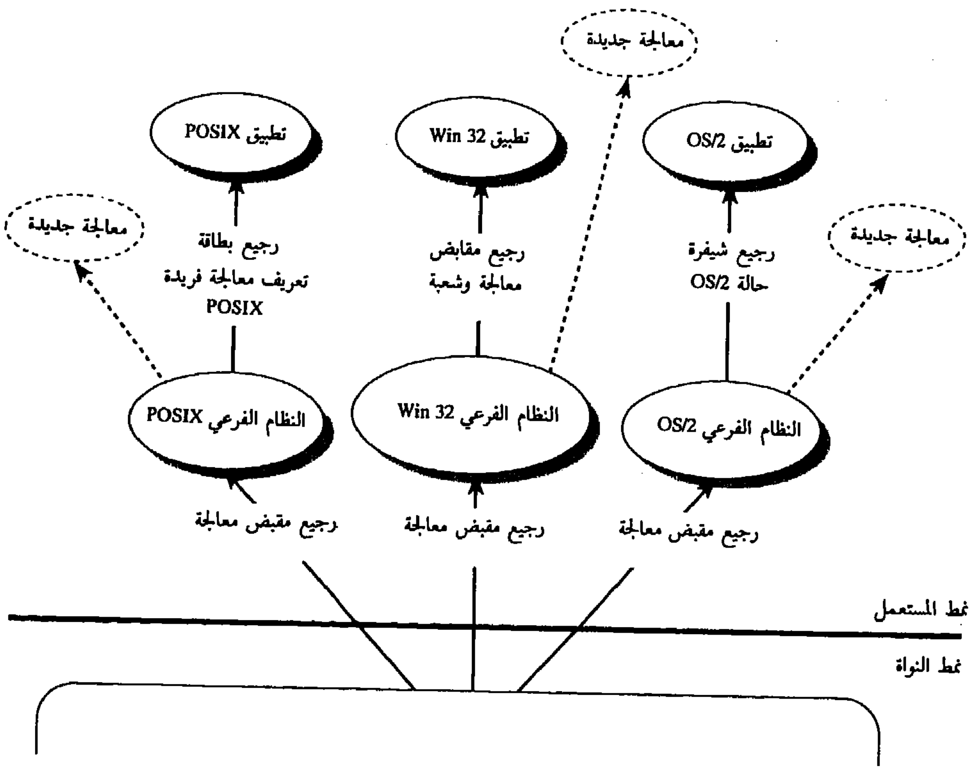


الشكل (4-11)
إنشاء معالجة

لاحظ أنه يتوجّب على النظام الفرعي لمحيط القيام بأعمال إضافيّة بعد إستلامه مقبض معالجة من برنامج إدارة المعالجة وقبل أن يرجع نتيجة إلى تطبيق المستضاف. فمثلاً، يستدعي النظام الفرعي برنامج إدارة المعالجة مجدداً لإنشاء شعبة للمعالجة الجديدة.

كما يُشاهد من الشكل (4-12)، ترجع محيطات نظام التشغيل المختلفة نتائج مختلفة عند إنشاء معالجة. وبشكل مشابه، تختلف أنظمة التشغيل في ناحية القواعد والمصطلحات المستعملة لإدارة المعالجات. إحدى الإختلافات الأساسيّة في محيطات نظام التشغيل المتوفّرة على Windows NT هي قدرتها على دعم المعالجات المتعدّدة الشعَب. فمثلاً، يتيح Win 32 و OS/2 عدّة شعَب لكل معالجة، بينما لا يتيح ذلك محيط POSIX و MS-DOS و Windows 16-bit.

مثال آخر على الإختلافات ضمن محيطات أنظمة التشغيل هي طريقة تعلّق كل معالجات النظام الفرعي مع بعضها البعض. فمثلاً POSIX و OS/2، ينظّمان معالجات المستضاف إلى

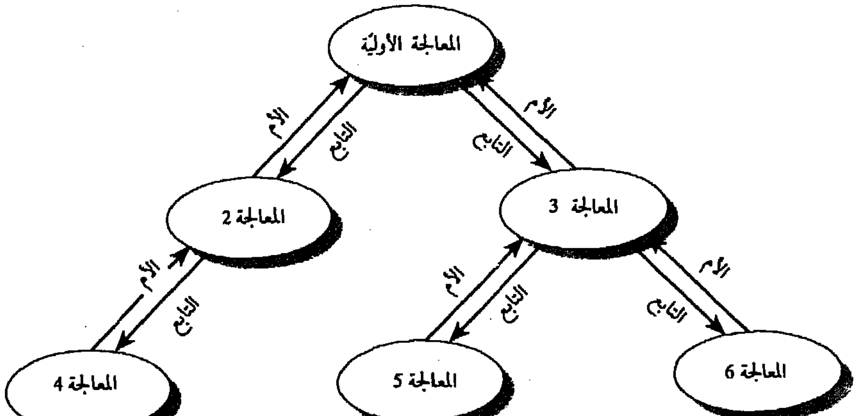


الشكل (4-12)
الرجيع من إنشاء المعالجة

تسلسلات هرميّة أو شجرة معالجة. وكل منها ينشىء معالجة أوليّة تنشىء ما يسمّى بالمعالجات التابعة. وبدورها تنشىء المعالجات التابعة معالجات تابعة لها. وباستثناء المعالجة الأوليّة، تحتوي كل معالجة على معالجة أمّ تأصّلت فيها موارد وخصائص معيّنة. يوضّح الشكل (4-13)، المبيّن على الصفحة التالية، هذه العلاقات.

يستعمل كل من POSIX و OS/2، العلاقات بين معالجات المستضاف لإدارتها. فمثلاً، عند إنهاء معالجة POSIX أو معالجة OS/2، يتعقّب نظام التشغيل هذه المعالجة وينهي كل المعالجات المنحدرة منها، إضافة لذلك، يحافظ نظام التشغيل POSIX على أنواع أخرى من العلاقات بين المعالجات، بما فيها مجموعات المعالجة (مجموعة من المعالجات المتعلّقة) والدورات (مجموعة من مجموعات المعالجة). تستخدم أنظمة POSIX ألسنيّة تحكُّم بالمعالجة مفصّلة متعلّقة بمجموعات المعالجة والدورات التي لا تحتوي على نسخ مطابقة في أنظمة تشغيل أخرى. يجب أن يتيح البرنامج التنفيذي NT لكل من أنظمته الفرعيّة للمحيط بدعم علاقات المعالجة التي تطلبها.

إضافة إلى الإختلافات في تجميع المعالجة وفي إستعمال المعالجات الاحاديّة الشعبة أو المعالجات المتعدّدة الشعَب، تختلف الأنظمة الفرعيّة للمحيط في القواعد التي تستعملها لإنشاء معالجات جديدة. يفصّل الجدول (4-4) بعض الإختلافات ضمن بنيات المعالجة لمحيطات نظام التشغيل الثلاث التي يدعمها النظام Windows NT.



الشكل (4-13)

التسلسل الهرمي للمعالجة POSIX أو OS/2

	Windows 32-bit	POSIX	OS/2
روتين API	Create Process ()	Fork ()	Dos ExecPgm()
التسلسل الهرمي للمعالجة	لا يحافظ على علاقات أم / تابع رسميّة	ينشىء معالجة جديدة كتابع للمستدعي	ينشىء معالجة جديدة كتابع للمستدعي
التأهّل	ينسخ إلى المعالجة التابع كل مقابض الكائن التي كانت مفتوحة مع صفة التأهيل	ينسخ واصفات ملفّ الأم إلى المعالجة التابع	ينسخ إلى المعالجة التابع كل مقابض ملف وأنبوب والإعلام الإشاري للأم والتي كانت مفتوحة مع حقوق التأصّل.
تحفيز فسحة العنوان	يحفّز فسحة عنوان المعالجة بواسطة برنامج قابل للتنفيذ	يحفّز فسحة عنوان تابع بنسخ فسحة عـنوان الأم	يحفّز فسحة عنوان تابع بواسطة برنامج قابل للتنفيذ
تعريف المعالجة	يرجّع مقبضاً إلى المعالجة الجديدة	يرجّع بطاقة تعريف المعالجة للتابع الجديد	يرجّع بطاقة تعريف المعالجة للتابع الجديد (إذا شغل
الشعَب	ينشىء شعبة واحدة ويدعم الشعَب المتعدّدة	ينشىء شعبة واحدة لكنه لا يدعم الشعَب المتعدّدة	التابع لا ينشىء شعبة واحدة ويدعم الشعَب المتعدّدة

الجدول (4-4)
ألسنيّة إنشاء المعالجة

كما يُشاهد، تختلف التسلسلات الهرميّة للمعالجة وتحفيز فسحة العنوان وتعريف المعالجة في المحيطات المختلفة. ورغم أن بعض هذه الإختلافات يبدو ثانوياً، يجب أن يدعم برنامج إدارة المعالجة كل المحيطات بالتساوي ويجب أن يتيح لبنيات المعالجة المختلفة التواجد دون أي تعارض. يشرح القسم التالي كيفيّة تحقيق ذلك.

2-3-4 بنية المعالجة المحليّة:

خلال تصميم بنية المعالجة NT المحليّة، أدرك المصمّمون أن توفير عدّة أنواع من بنيات المعالجة في نظام التشغيل المرجعي سيؤدّي إلى نظام معقّد جداً. ولحسن الحظ، فإن معظم التفاصيل المتعلّقة ببنية المعالجة ليست أساسيّة لتشغيل نظام التشغيل الأساسي. يمكن إستخدام بنيات المعالجة في الأنظمة الفرعيّة للمحيط في نمط المستعمل خارج البرنامج التنفيذي NT. ولتحقيق ذلك، لا تفرض بنية المعالجة للبرنامج التنفيذي أية مجموعة من القواعد التي يمكن أن تمنع مجموعة أخرى. وإنما فهي تزوّد مجموعة مرجعيّة من الآليات التي تستطيع الأنظمة الفرعيّة إستعمالها كأساس لاستخدام بنيات المعالجة الخاصة بها. وكمثال لهـذه الطريقـة، يفصّل الجدول (5-4)، المبينّ على الصفحة التالية، قواعد إنشاء المعالجة المرنة للبرنامج التنفيذي.

يعتبر البرنامج التنفيذي عمليّة إنشاء المعالجة على أنها إنشاء كائن ــ وليس شيئاً آخر. وعند إنتهاء برنامج إدارة المعالجة من إنشاء معالجة جديدة، فإنه يرجع مقبض المعالجة الجديدة إلى النظام الفرعي للمحيط. يكون النظام الفرعي مسؤولاً عن إستدعاء برنامج إدارة المعالجة لإنشاء شعبة في المعالجة.

لا يسجّل برنامج إدارة المعالجة NT المعالجات المنشأة من قبَل المعالجات الأخرى. لذلك، ولمحاكاة علاقات المعالجة الفريدة المطلوبة من قبَل تطبيقاتها، يحافظ كل نظام فرعي لمحيط على سجلّات المعالجات المستضاف الذي أنشأها والعلاقات الموجودة بينها. تصف الأقسام التالية

	NT
روتين API	Nt Greate Process ()
التسلسل الهرمي للمعالجة	ينشيء معالجة جديدة كنظير مستقلّ لمستدعي ويرجّع مقبض كائن.
التأصُّل	يحدّد المستدعي معالجة أم تتأصّل منها المعالجة الجديدة مقابض الكائن التي كانت مفتوحة في صفة التأصُّل.
تحفيز فسحة العنوان	يحفّز فسحة عنوان معالجة جديدة بواسطة برنامج قابل للتنفيذ أو كنسخة عن فسحة عنوان الأم.
تعريف المعالجة	يرجّع مقبض كائن NT لمعالجة جديدة.
الشعَب	لا ينشيء شعبة في المعالجة الجديدة تلقائياً، لكنّه يدعم الشعَب المتعدّدة.

الجدول (4-5)
ألسنيّة إنشاء معالجة NT محليّة

بعض البرامج الخدماتيّة التي يوفّرها البرنامج التنفيذي NT للأنظمة الفرعيّة لإدارة مستضافاتها.

4-3-2-1 إدارة معالجات المستضاف:

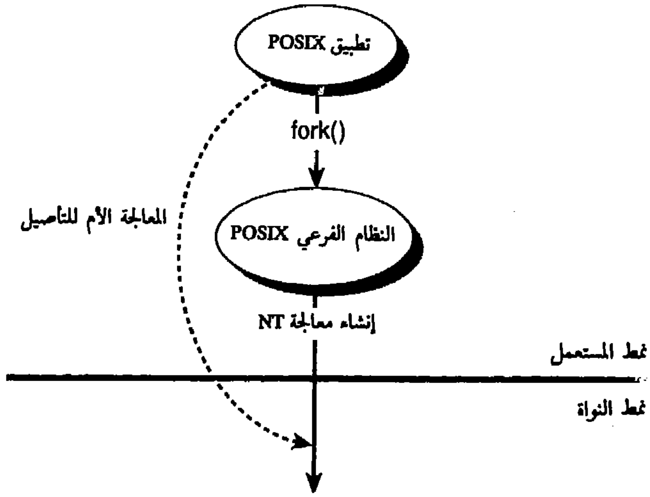
يجب توفير مجموعة دنيا من الموارد لمعالجة NT محليّة، إضافة إلى شعبة، قبل تمكّنها من القيام بأي عمل. وإذا راجعتَ الشكل (4-2) على الصفحة 92، ستشاهد أن المعالجة تحتوي على صفة وصول ومحتويات فسحة العنوان ومقابض إلى الكائنات. فهذه الموارد، إضافة للحصص المحدّدة للمعالجة والضوابط الأخرى، متأصّلة كلياً أو جزئياً من معالجة أخرى، «معالجة الأم».

تحصر علاقة الإقتباس التعبير «معالجة الأم» بسبب نظريّة الأم المعيّنة في البرنامج التنفيذي NT. راجع الرسم التوضيحي في الشكل (4-14).

في هذا الشكل، يستدعي تطبيق POSIX النظام الفرعي POSIX لإنشاء معالجة POSIX جديدة. ويستدعي النظام الفرعي، الذي هو معالجة أيضاً، البرنامج التنفيذي NT لإنشاء معالجة محليّة. ولأن النظام الفرعي POSIX يعمل نيابة عن تطبيق المستضاف، يجب أن تتأصّل المعالجة الجديدة مواردها من المستضاف وليس من النظام الفرعي. وهذا الأمر صحيح بالنسبة لتطبيقات Win 32 و OS/2 التي تنشىء معالجات جديدة. وللإتاحة للأنظمة الفرعيّة للمحيط لمحاكاة ألسنيّة تأصُل المعالجة المطلوبة من قبَل تطبيقاتها، تتيح خدمة إنشاء المعالجة في البرنامج التنفيذي NT لمستدعي (النظام الفرعي في هذه الحالة) تحديداً إختيارياً أم المعالجة الجديدة.

تتأصّل المعالجة NT الجديدة صفة الوصول وحدود الحصّة والأولويّة المرجعيّة وصلة المعالج المفترضة. كذلك، تتأصّل المعالجة أية مقابض في جدول كائنات الأم التي كانت مفتوحة مع تسمية التأصيل. يمكن أيضاً تأصيل فسحة عنوان الأم إذا طلبها النظام الفرعي. يستعمل النظام الفرعي POSIX هذه المزيّة لمحاكاة روتين API Fork () في POSIX بينما يحدّد النظام الفرعي Win 32 و OS/2 رسماً قابلاً للتنفيذ لتحميله في فسحة عنوان المعالجة الجديدة.

وقبل التنفيذ، يجب توفير شعبة للمعالجة الجديدة. فبالنسبة لتطبيقات Win 32 أو OS/2، لا يعتبر إنشاء شعبة عمليّة مستقلّة عن إنشاء معالجة جديدة. تتوقّع تطبيقات Win 32 و OS/2 أن تكون الشعبة موجودة عندما يرجع روتين إنشاء المعالجة. ولا يتمّ إنشاء الشعبة تلقائياً في NT،



الشكل (4-14)
تعيين معالجة الأم

لذلك يجب أن تستدعي الأنظمة الفرعيّة برنامج إدارة المعالجة مجدّداً لإنشاء شعبة في المعالجة الجديدة، وعند إنشاء شعبة، يتيح برنامج إدارة المعالجة للنظام الفرعي تحديد معالجة تنسب إليها الشعبة الجديدة. وهذا يتيح للنظام الفرعي Win 32، مثلاً، إنشاء معالجة لإحدى مستضافاتها ثم وضع شعبة في فسحة عنوان المستضاف. يبدأ تنفيذ الشعبة الجديدة عند الأولويّة المرجعيّة لمعالجة المستضاف، على مجموعة المعالجات المسردة في قائمة صلة معالج المستضاف ويظلّ تقييدات صفة الوصول المستضاف.

إضافة لإنشاء المعالجات والشعَب نيابة عن المعالجات الأخرى، يوفّر برنامج إدارة المعالجة NT برامج خدماتيّة تتيح لنظام فرعي إلحاق فسحة عنوان مستضاف والقراءة والكتابة إليه وتخصيص وإخلاء الذاكرة الظاهريّة للمستضاف وفي تعليق شعب مستضاف وتعديل حالات تنفيذ الشعَب وإعادة بدء تشغيلها.

يستطيع النظام الفرعي إستنساخ مقابض الكائن من الجدول الخاص بكائناته في جدول كائنات المستضاف. إضافة لذلك، يستطيع إنهاء شعَب مستضاف أو معالجة مستضاف.

توفّر هذه القدرات القوية للأنظمة الفرعيّة في نمط المستعمل التحكّم الذي يكون محدّداً في معظم أنظمة التشغيل لشيفرة نظام التشغيل في نمط النواة. وهي توفّر الحريّة للأنظمة الفرعيّة للمحيط في التحكُّم بتطبيقات المستضاف وفي تحقيق محيط نظام تشغيل لها يختلف عن المحيط المحلي للبرنامج التنفيذي NT.

2-2-3-4 منع سوء الإستعمال:

إن إنشاء معالجات وشعَب نيابة عن معالجة أخرى والقراءة والكتابة إلى الذاكرة الظاهريّة لمعالجة أخرى والتحكُّم بشعَب معالجة أخرى هي عمليات لا يجب إستعمالها عشوائياً. ولمنع سوء الإستعمال، يضمن نظام الأمان في البرنامج التنفيذي NT (وخاصة آليات حماية الكائنات) التحكُّم بمثل هذه العمليات.

فعند مستواها المرجعي، تكون الأنظمة الفرعيّة للمحيط NT معالجات عاديّة بسيطة. ومثل المعالجات الأخرى، فهي تحدّد حقوق الوصول الممنوحة إلى المعالجات التي تنشئها. ولأن كل المعالجات في نمط المستعمل تنشأ بواسطة الأنظمة الفرعيّة للمحيط، تتحكَّم الأنظمة الفرعيّة بكل أعمال معالجات المستعمل في النظام.

على سبيل المثال، عند إنشاء معالجة مستضاف، يمنع نظام فرعي عن المستضاف قدرة تجاوز النظام الفرعي وإنهاء نفسه باستدعاء خدمة NT محليّة. ففي هذه الحالة، يمكن أن تترك

المعالجة بنيات البيانات العامة للنظام الفرعي في حالة غير متماسكة حيث يمكن أن تقاطع المعالجات الأخرى الموجودة بظل تحكّم ذلك المحيط. يمنع النظام الفرعي ذلك عن طريق منع الوصول لحذف المعالجة الجديدة إلى كائن المعالجة الخاص به في قائمة التحكّم بوصول الكائن (ACL). (راجع الفصل الثالث «برنامج إدارة الكائنات وأمان الكائن»). ودون الوصول للحذف، لا تستطيع المعالجة فتح مقبض كائن يتيح إنهاء نفسها ولن يتيح نظام الأمان لخدمة الإنهاء بالنجاح ما لم يتوفّر لها مقبض صالح.

بسبب طريقة عمل أمان الكائن، وما لم يحدّد نظام فرعي قدرة لمعالجة مستضاف، لا يستطيع المستضاف الحصول عليه. وهكذا لا يحتاج مصمّم نظام فرعي للتفكير عن الأشياء السيّئة التي يمكن أن تقوم بها معالجة مستعمل وطريقة منعها. لكن وبدلاً من ذلك، يكفي تحديد ما يمكن أن تقوم به المعالجة ومنح المستضاف هذه القدرات. وهذا يعني أنه لا يمكن لأية معالجة في نمط المستعمل عادية النجاح في إستدعاء خدمات NT محليّة. تستطيع المعالجة في نمط المستعمل إستدعاء فقط روتينات API المتوفّرة من قبَل النظام الفرعي الذي أنشأها.

تمنع هذه الآليات أيضاً المعالجات في نمط المستعمل من إنهاء المعالجات الأخرى ومناولتها. تستطيع المعالجة إستدعاء فقط روتينات API المتوفّرة في محيطها (Win 32 أو POSIX على سبيل المثال) لمناولة معالجة أخرى. إضافة لذلك، تقيّد قدرة معالجة على إستدعاء هذه الخدمات بواسطة حقوق الوصول إلى الكائنات المحليّة التي تؤثّر عليها. ومرّة أخرى، وعن طريق عدم منح حقوق الوصول لمعالجة مستضاف إلى كائنات محليّة، يستطيع نظام فرعي لمحيط منع المستضاف من سوء التصرُّف. وان تصميم الأمان في Windows NT الذي يعمل دون توقُّف في الخلفيّة، يضمن ذلك.

4-4 بإختصار:

المعالجات هي أقسام أساسيّة من العمل والموارد ضمن النظام Windows NT. وهي تتيح لنظام التشغيل تقسيم عمله إلى وحدات وظيفيّة لتحقيق الإستعمال الكافي للمعالج. يقسّم النظام Windows NT المعالجات إلى وحدات قابلة للتنفيذ تسمى شعَب. وتتيح الشعَب لمعالجة واحدة تنفيذ أجزاء مختلفة من برنامجها يتزامن وتحقيق إستعمال أفضل للمعالج وخاصة على الحواسيب المتعدّدة المعالجات. تتألف المعالجة من فسحة عنوان ومجموعة موارد تتشاركها كل الشعَب عند التنفيذ.

إن بنية معالجة البرنامج التنفيذي NT هي أوليّة ومرنة تتيح للأنظمة الفرعيّة للمحيط

إنشاء ألسنيّة تحتاجها لدعم مستضافاتها. وهي تستطيع إنشاء تسلسل هرمي للمعالجة وإستخدام تأصل المعالجة وتحفيز فسحة العنوان لمستضافاتها كما تتلاءم. ويمكنها التحكُّم أيضاً بمعالجات المستضاف بطرُق أخرى، عن طريق تنفيذ أعمال نيابة عن مستضافاتها وبقراءة الذاكرة الظاهريّة للمستضاف والكتابة إليها. تراقب كل هذه القدرات بواسطة نظام الأمان الذي يدقّق بحقوق الوصول التي تحتويها معالجة على الكائنات التي تحاول مناولتها قبل الإتاحة لمثل هذه العمليات.

يركّز الفصل التالي بتفصيل أكبر على بنية الأنظمة الفرعيّة Windows NT وعلى البرنامج الخدماتي لتمرير الرسائل LPC الذي يتيح لمعالجات المستضاف وللأنظمة الفرعيّة الإتصال مع بعضها البعض.

5

النظام Windows والأنظمة الفرعيّة المحميّة

خلال تطوير النظام Windows NT، أُطلق عليه عدّة تسميات كـ «أمّ كل أنظمة التشغيل» وكـ «أفعى متعدّدة الرؤوس». وكانت أنظمته الفرعيّة المحميّة، وخاصة أنظمته الفرعيّة للمحيط، السبب لهذه التسميات.

وكما يشير الإسم Windows NT، يزود إصدار Windows من 32 بت نظام التداخل مع المستعمل للبرنامج التنفيذي NT. وبالنسبة للمستعمل، فإن Windows NT يبدو مثل Windows على MS-DOS. ولكن للمبرمج، إتخذ Windows NT عدّة شخصيات.

إن فكرة قيام نظام تشغيل واحدة بتشغيل أنواع مختلفة من البرامج ليس بالأمر الجديد. فقد استعار النظام Windows NT هذه الفكرة من نظام التشغيل Mach، الذي صمّم لدعم إصدارات مختلفة وغير متوافقة من تداخلات البرمجة التطبيقيّة (APIs) من UNIX ضمن نفس نظام التشغيل. وقد حقّق نظام التشغيل Mach ذلك باستخدام محيطات API مختلفة كمعالجات ملقّم في نمط المستعمل. يستعمل النظام Windows NT نفس الطريقة لتحقيق الأهداف المختلفة.

ومن المرحلة الأولى لتطوير Windows NT، ركّز المصمّمون على تمديد تداخلات API الموجودة في Microsoft وليس إستبدالها، بحيث تستمرّ التطبيقات الموجودة بالإشتغال خلال تطوير تطبيقات جديدة. أما أهم نتيجة لهذا الجهد فهو نظام التداخل 32-bit Windows الذي يسمّى Win 32 API. يتيح Win 32 API للتطبيقات إستعمال قدرات نظام التشغيل المعقّدة التي لم تكن متوفّرة في 16-bit Windows API.

إضافة لتشغيل تطبيقات Win 32، يشغّل النظام Windows NT تطبيقات MS-DOS و 16-bit Windows الموجودة وكذلك العديد من تطبيقات OS/2 وكل إمتثالات (IEEE 1003.1 POSIX).

لتحقيق هذه المرونة فإن البرنامج التنفيذي عام. وهو يتناول إنشاء المعالجة بمستوى

منخفض والشعبة وجدولة الشعَب وإدارة الذاكرة ومناولة المقاطعة والدخل / الخرج مع الإعتماد على ملقّمات نمط المستعمل لتوفير التداخل التخطيطي مع المستعمل والمزايا الأخرى التي تتوقّعها التطبيقات والمستعملين. باستعمال خدمات النظام NT كمرجع، تستخدم ملقّمات نمط المستعمل المستقلّة روتينات API في Win 32 و 16-bit Windows و MS-DOS و POSIX و OS/2. ويمكن أن يتواجد أي عدد من روتينات API ومحيطات تنفيذ التطبيقات في النظام Windows NT.

تسمّى الملقّمات التي توفّر محيطات API، الأنظمة الفرعيّة المحميّة وبالتحديد الأنظمة الفرعيّة للمحيط. ورغم أن كل نظام فرعي مصمّم من قبَل مصمّم واحد أو عدّة مصمّمين مختلفين، صمّمت طريقة المستضاف / الملقّم الإجماليّة وأول نظامين فرعيين للمحيط (إصدار OS/2 و POSIX) من قبَل Steve Wood و Mark Lucovsky. أما الأهداف العامة التي وصفت للأنظمة الفرعيّة للمحيط في Windows NT فهي:

- جعل كل نظام فرعي قوي بحيث لا تستطيع تطبيقات المستضاف التأثير سلباً على بعضها البعض أو على النظام الفرعي ككلّ. كذلك التأكّد من عدم تمكّن نظام فرعي واحد من التأثير إعتباطياً على نظام فرعي آخر أو على تطبيقات المستضاف العائدة له.

- التأكّد من مقارنة أداء كل محيط بشكل موجب مع أداء نظام التشغيل الذي يحاكيه. فمثلاً، يجب أن تشتغل تطبيقات 16-bit Windows على Windows NT بنفس السرعة التي تشتغل فيها على 16-bit Windows. وبشكل مشابه، يجب مقارنة تنفيذ تطبيقات MS-DOS على Windows NT بشكل إيجابي مع تنفيذها على نظام MS-DOS محلي.

- التأكّد من ملاءمة كل نظام فرعي مع متطلّبات الحكومة الأميركيّة لناحية محيط نظام تشغيل آمن. وهذا يتضمّن التدريع الكامل لكل ذاكرة معالجة من المعالجات الأخرى والتحكّم بكل وصول لمعالجة مستضاف إلى موارد النظام الفرعي وإلى موارد المستضافات الأخرى.

- الإتاحة للأنظمة العمل داخلياً عند توقّع المستعمل قيامها بذلك. فمثلاً، يجب أن تتمكّن تطبيقات 16-bit و 32-bit من Windows من تمرير البيانات إلى بعضها البعض عبر الحافظة Clipboard أو للإتصال باستعمال تبادل البيانات الديناميكي (DDE) أو ربط الكائنات وتضمينها (OLE). ويجب على محلّل أمر في نمط حرف واحد أن ينفّذ برامج MS-DOS و OS/2 أو POSIX والإتاحة بإعادة توجيه الدخل / الخرج بينها عند الضرورة. ويجب أن تتمكّن كل التطبيقات في نمط الحرف من إرسال خرجها إلى دخل / خرج قياسي وجعل النظام يعرضه تلقائياً في إطار.

لا يصف هذا الفصل كل نظام فرعي بتفصيل. لكنه يركّز على محيط Win 32 ومحيط

16-bit Windows ومحيط MS-DOS. إن النظام الفرعي لمحيط Win 32 هو مكوّن دقيق للنظام Windows NT وهو يوفّر التداخل مع المستعمل والبرمجة في النظام. ومحيطات 16-bit Windows و MS-DOS مهمّة لتوافقها مع التطبيقات الموجودة.

يصف القسم الأول من هذا الفصل نموذج المستضاف / الملقّم في النظام Windows NT وكيفيّة إنتقائه. ويشير القسم الثاني إلى وسائل تفاعل الأنظمة الفرعيّة، ويركّز القسم الثالث على النظام الفرعي Win 32. يتبع ذلك قسم عن محيطات MS-DOS و 16-bit Windows وأخيراً شرح عن البرنامج الخدماتي لتمرير الرسائل في النظام.

1-5 نظرة شاملة حول الأنظمة الفرعيّة المحميّة:

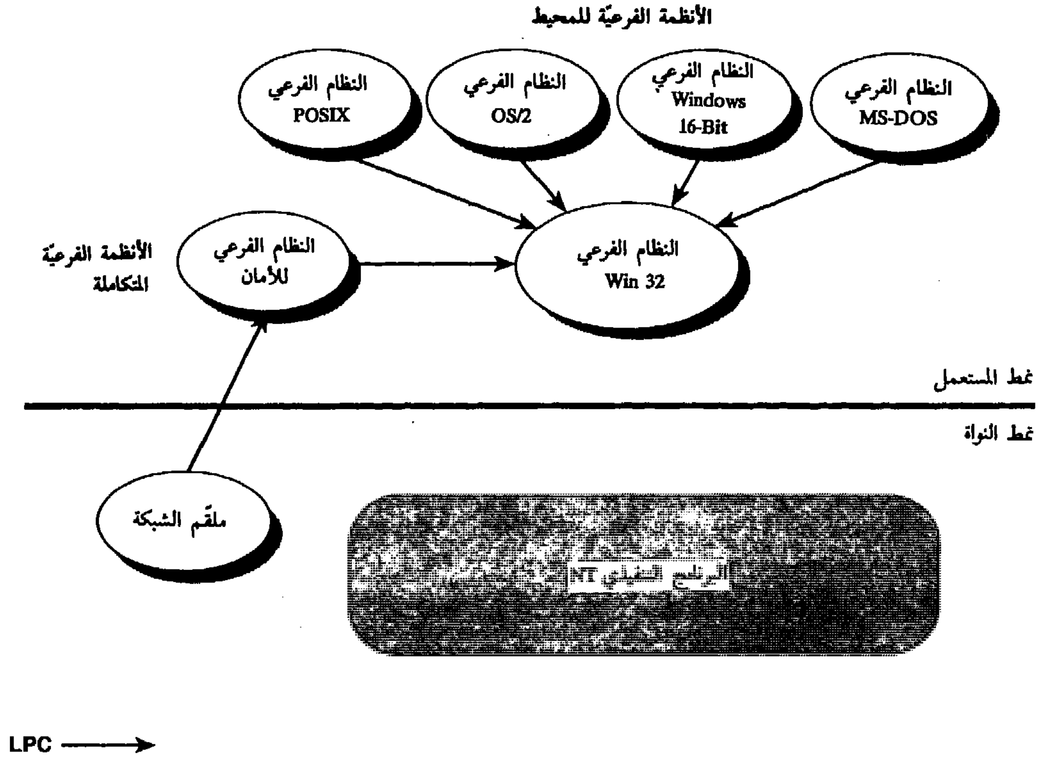
الأنظمة الفرعيّة المحميّة في النظام Windows NT هي معالجات ملقّم في نمط المستعمل يتمّ إنشاءها من قبَل النظام Windows NT عند إستنهاض نظام التشغيل. وبعد إنشائها، فإنها تعمل باستمرار، حيث تستجيب للرسائل المرسلة إليها من قبَل معالجات التطبيقات ومن الأنظمة الفرعيّة الأخرى. تستخدم بعض الأنظمة الفرعيّة للمحيط روتينات API في نظام التشغيل بينما ينفّذ نوع آخر من الأنظمة الفرعيّة والذي يسمّى النظام الفرعي المتكامل، مهام نظام التشغيل الضروريّة. يستخدم معظم نظام الأمان في Windows NT كنظام فرعي متكامل، ويمكن أيضاً إستعمال ملقّمات الشبكة كأنظمة فرعيّة متكاملة.

ينفّذ كل نظام فرعي محمي عمله في نمط المستعمل، حيث يستدعي خدمات نظام البرنامج التنفيذي NT لدعم نظام التشغيل في نمط النواة. يمكن أن تشتغل ملقّمات الشبكة إما في نمط المستعمل أو في نمط النواة، وفقاً لتصميمها. تبينّ الأنظمة الفرعيّة في Windows NT في الشكل (1-5) على الصفحة التالية.

تتّصل الأنظمة الفرعيّة عن طريق تمرير الرسائل إلى بعضها البعض. فعندما يستدعي تطبيق مستعمل روتين API، على سبيل المثال، يستلم النظام الفرعي للمحيط الذي يوفّر الروتين رسالة ويستخدمها باستدعاء خدمات النظام NT أو بتمرير الرسائل إلى الأنظمة الفرعيّة الأخرى. وعند الإنتهاء، يرسل النظام الفرعي للمحيط رسالة تحتوي القيم الرجعيّة إلى التطبيق. تكون عمليّة تمرير الرسائل والنشاطات الأخرى للأنظمة الفرعيّة المحميّة غير مرئيّة للمستعمل.

إن الذي يحافظ على تماسك نموذج المستضاف / الملقّم في Windows NT هو ما يُعرَف

بالبرنامج الخدماتي لاستدعاء إجراء محلي (LPC). وهو برنامج خدماتي لتمرير الرسائل حيث تطلب المستضافات بواسطتها الملقّمات وحيث تستجيب الملقّمات. إن إستدعاء LPC هو إصدار محلي مستمثل لبرنامج خدماتي لتحرير الرسائل أكثر شمولية يسمى إستدعاء إجراء بعيد (RPC) الذي يستعمل للإتصال ضمن معالجات المستضاف والملقّم المتواجدة على حواسيب مختلفة في شبكة.



الشكل (5-1)
الأنظمة الفرعيّة المحميّة في Windows NT

يزوّد البرنامج الخدماتي LPC عدّة طرق لتمرير البيانات بين المستضافات والملقّمات، فواحدة تُستعمل لإرسال الرسائل القصيرة والأخرى تُستعمل لإرسال رسائل طويلة والثالثة تُستمثل للإستعمال من قبَل النظام الفرعي Win 32. ينشىء كل نظام فرعي منفذ ــ وهو عبارة عن قناة إتصال تتّصل عبرها المعالجات الأخرى مع النظام الفرعي، تستخدم المنافذ ككائنات في البرنامج التنفيذي NT.

يعالج هذا القسم نموذج المستضاف / الملقّم في Windows NT من وجهة النظر التاريخيّة ـ أي كيفيّة تطوُّر النموذج خلال المراحل الأولى لتصميم النظام Windows NT. ويركّز القسم الفرعي الأول على كيفيّة قيام نموذج المستضاف / الملقّم بمساعدة النظام Windows NT لتحقيق أهدافه التصميميّة. ويعالج القسم الفرعي الثاني إعتباراً رئيسياً عند إستعمال نموذج المستضاف / الملقّم: أداء النظام.

1-1-5 لماذا إستعمال نموذج مستضاف / ملقّم؟

إن إستخدام أجزاء من نظام التشغيل في ملقّمات نمط المستعمل هو جزء مهمّ في تصميم النظام Windows NT وبالتالي يؤثّر إلى حدٍّ بعيد على كيفيّة عمل النظام، ورغم عدم كون طريقة المستضاف / الملقّم خاصة بتصميم نظام التشغيل إلا أنها تتّفق مع الطرق المستعملة. وحتى مؤخراً، كان يتمُّ إنشاء معظم أنظمة التشغيل حول فيما إذا كان نموذجاً أحادي الطبقة أو نموذجاً متعدّد الطبقات. وفي هذين النموذجين، يشتغل نظام التشغيل في نمط معالج بأفضليّة (أو ربما بأكثر من نمط واحد بأفضليّة) والذي يفرّق نظام التشغيل عن شيفرة التطبيق دون أفضلية. إضافة لذلك، ومن نموذج أحادي الطبقة أو بطبقات، لا يشتغل أي جزء من نظام التشغيل في معالجة لوحده. لكن وعوضاً عن ذلك، ينشىء نظام التشغيل معالجات في نمط المستعمل لتشغيل التطبيقات وتنفّذ شيفرة نظام التشغيل نيابة عن هذه التطبيقات عندما تستدعي خدمات النظام أو عند حصول مقاطعات خارجيّة.

من الناحية المقابلة، يتألف النظام Windows NT من جزئين: قسم نظام تشغيل تقليدي ومجموعة من «تطبيقات» نمط المستعمل الذي تنفذ مهام نظام التشغيل. إن «التطبيقات» هي أنظمة فرعيّة محميّة في Windows NT. ومثل التطبيقات العاديّة، فإنها تنفّذ ضمن معالجة مع فسحة عنوان خاص. ويجدول مكوِّن النواة في البرنامج التنفيذي NT شعبها على المعالجات تماماً كما يفعل للتطبيقات الأخرى.

يستعمل النظام Windows NT الأنظمة الفرعيّة المحميّة لتحقيق الأهداف التالية:

- توفير تداخلات البرمجة التطبيقيّة المتعدّدة (APIs) مع إبقاء شيفرة نظام التشغيل المرجعي (البرنامج التنفيذي NT) سهلة وقابلة للصيانة.
- حماية نظام التشغيل المرجعي من التغييرات في روتينات API المتوفّرة أو من التمديدات عليها.
- دمج البيانات العامة المطلوبة من قبل كل روتين API وفي نفس الوقت، فصل البيانات العامة المطلوبة من قبل روتين API واحد عن تلك المطلوبة من قبل روتينات API الأخرى.

■ حماية كل محيط API من التطبيقات ومن بعضها البعض وحماية نظام التشغيل المرجعي من المحيطات المختلفة.

■ إتاحة تمديد نظام التشغيل مع روتينات API الجديدة في المستقبل.

لقد كان البند الأول في هذه القائمة، توفير روتينات API متعدّدة، هدف مهمّ للنظام Windows NT وقد شكّلت هذه معضلة لم يكن حلّها واضحاً. إن هذا القسم الفرعي يطرح المناقشات التي دارت حول كيفيّة تحقيق هذا الهدف ويذكّر لماذا تمّ إختيار نموذج المستضاف / الملقّم.

1-1-1-5 توفير محيطات متعدّدة:

كما شرح في الفصل الأول «المهمّة»، فرضت متطلبات السوق الأصلية على NT إمداد تداخلات برمجة لتطبيقات OS/2 و POSIX في المرحلة الأولى ثم إتاحة إضافة روتينات API أخرى في المستقبل. ولقد كان هذا هو الإفتراض الذي إعتمده أفراد الفريق عندما بدأوا تصميم بنية NT (كما كان يسمّى النظام في العام 1988).

ولقد أدرك الفريق لاحقاً وخلال تدقيقه بهذا الموضوع في العام 1988، أنه لم يكن كافياً إستخدام روتينات API متعدّدة لأنها غير موجودة في فراغ هوائي. فمثلاً، فإن تطبيق OS/2 الذي يستدعي الروتين () DosExecPgm لإنـشاء معالجة وتشغيل برنامج، يتوقع من الروتين القيام بعمله وإرجاع القيم الصحيحة ـ إضافة لذلك، وهذا هو صلب الموضوع، يتوقّع التطبيق أيضاً من الروتين إنشاء معالجة تتصرّف تماماً كما تفعل على نظام OS/2. ونفس الشيء الصحيح لتطبيقات Windows و MS-DOS و POSIX، أي يجب على روتين API ومحيط التنفيذ الأساسي العائد له أن يتوافقا كلياً مع المحيط المحلّي للتطبيق. يتّصف كل نظام تشغيل محلي ببنية معالجة مختلفة وإدارة ذاكرة مختلفة ومناولة إستثناء وخطأ مختلف وآليات حماية موارد مختلفة وألسنيّة مختلفة للوصول إلى الملفّات والدخل / الخرج. فكيف يمكن توفير محيط تنفيذ متوافق مع العديد من أنظمة التشغيل المختلفة؟ لقد كان ذلك التحدّي الأكبر في تصميم النظام Windows NT.

بدأ الفريق بمراجعة روتينات API في OS/2 و POSIX ومحاولة إنتقاء بنية للنظام NT تستوعب كلاهما، وقد أضيف هدف آخر يضمن إمكانيّة توفير النظام روتينات API أخرى غير محددة بعد للمدوديّة المستقبليّة. (ولقد كانت روتينات API للأنظمة Windows و MS-DOS هي الخيار الطبيعي).

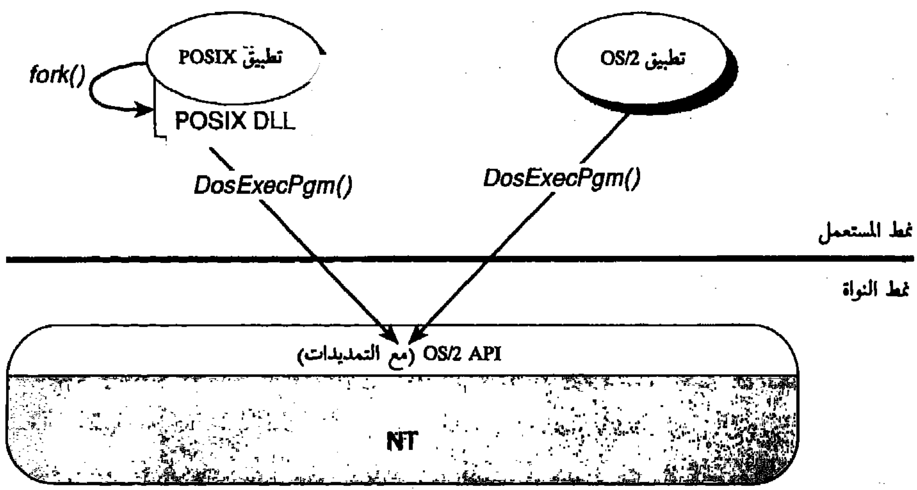
تمركزت الفكرة الأولى حول جعل النظام NT نظام تشغيل بطبقات قياسي يعرض روتين API منتقىً واحداً كخدمات نظام محليّة. ولقد كان روتين API في OS/2 الخيار الأول في ذلك

الوقت. وقد تتواجد محيطات POSIX والأخرى المستقبليّة كتداخـلات وقت التشغيل التي إستدعت خدمات مثل OS/2 لتنفيذ عملها (راجع الشكل (5-2)).

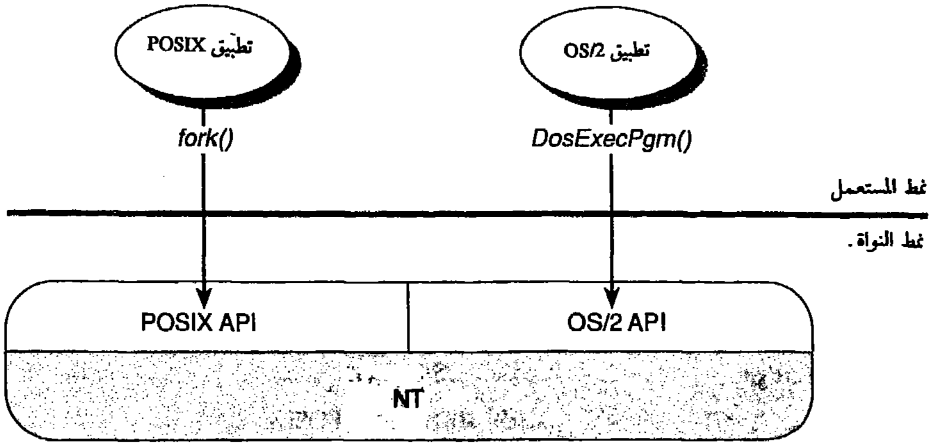
وعند الإمعان بهذا الخيار، أصبح من الواضح أن تصرّف كل روتينات OS/2 API تختلف عن تصرّف روتينات POSIX التي يجب أن تحتاج روتينات OS/2 لتغييرها. فمثلاً، قد يحتاج روتين إنشاء معالجة في OS/2، DosExecPgm ()، لدعم خيار نسخ فسحة عنوان من معالجة أم إلى معالجة تابع. (لحفز التصرُّف العادي للروتين ذاكرة معالجة التابع بصورة قابلة للتنفيذ). لكن الأمر الصعب كان ألسنيّة OS/2 و POSIX التي تعذّر فصلها عن بعضها البعض أو عن NT بشكل جيّد. فمثلاً، إذا استدعى تطبيق OS/2 متعدّد الشعّب الروتين Dos Exec Pgm () عن طريق الخطأ وحدّد الخيار POSIX، يخفق نظام التشغيل لأن النظام NT لا يتوقع أن تحتوي معالجة POSIX أكثر من شعبة واحدة. لذلك، رفض خيار التصميم الأول لأنه لا يضمن نظام تشغيل قوي وقابل للصيانة أو مدودي.

خيار التصميم الثاني كان في جعل NT نظام API مزدوجاً، بوضع روتينات API لكل من OS/2 و POSIX مباشرة في NT، والتشغيل في نمط النواة كما يوضح ذلك الشكل (5-3).

بواسطة هذا الخيار، لن تتمكّن طبقات API من إدارة البيانات العامة أو تعقُّب حالة المعالجات والذاكرة وما شابه، لكنها تستدعي الطبقة NT التي تستخدم محيط نظام تشغيل عام يدعم ألسّنيّتي OS/2 و POSIX.



الشكل (5-2)
إستعمال روتينات API المتعدّدة (الخيار الأول)



الشكل (3-5)
إستعمال طبقات API متعدّدة (الخيار الثاني)

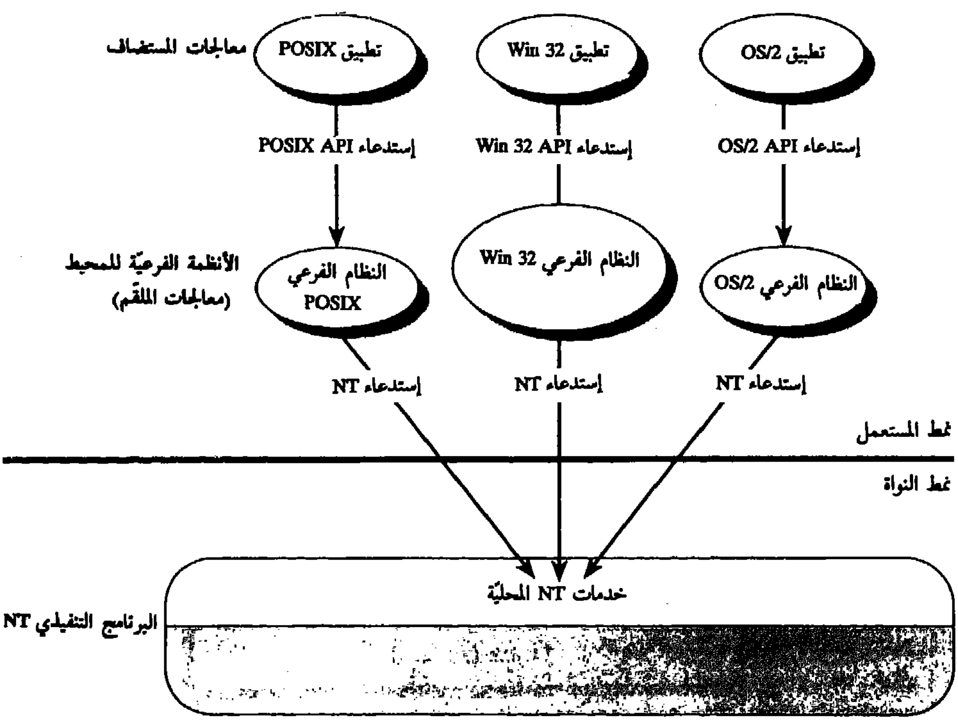
وبالتدقيق بهذه الفكرة بإمعان، كان من الواضح أنها تحسين ضئيل على الخيار السابق، ورغم أنه يتعذّر على برنامج في نمط المستعمل إستدعاء NT ببارامترات خطيرة، فإن NT نفسه يحتاج لدعم محيطي تنفيذاً لتطبيقات غير متوافقين. فمثلاً، يحتاج روتين إنشاء المعالجة إلى إستخدام ألسُنيّتي OS/2 و POSIX واستعمال علم يشير إلى نوع المعالجة التي يريد إنشاءها المستدعي. وبشكل مشابه، عند إنتهاء معالجة، يحتاج NT لتحديد أنها معالجة OS/2 أنشئت بواسطة استعمال الخيار EXEC_SYNC. في هذه الحالة، يمكن معاودة إستعمال معرّف المعالجة من قبَل معالجة أخرى. فإذا كانت معالجة POSIX وإذا لم تكن المعالجة الأم المعالجة الأوليّة، يرسلها NT كإشارة شكل POSIX. وإذا كانت المعالجة رئيس مجموعة دورة، يولّد البرنامج التنفيذي إشارة تعليق إلى كل أفراد مجموعة الدورة بإستعمال نفس طرف التحكّم وقد يخلي طرف التحكّم وهكذا فهي عملية فوضويّة ومعقّدة.

لم تكن بنية المعالجة منطقة الصعوبة الوحيدة، فقد ظهرت المشاكل نتيجة للإختلافات الدقيقة بين OS/2 و POSIX في عدّة مجالات بما فيها الموقتات ونسق تاريخ اليوم وإقفال الملفّات والأنابيب والمناولة الإستثنائيّة والأخرى. ولدعم هذه الإختلافات، يجب أن تنقل كل معالجة وشعبة حقيبة تعرّف خصائصها وتشير إلى الجداول الخاصة لمتابعة تعقّب التوليفات والأعمال الممكنة. غير أنه قد يصعب إستخدام بعض الوظائف البسيطة، مثل إنتظار معالجة تابع (مشتركة لكل من OS/2 و POSIX)، لأن NT يحتاج لإدارة حالتين مختلفتين (دون ذكر الحالات المستقبليّة). وكما ذكر Mark Lucovsky «إن قائمة تداخلات الأسلاك التافهة المطلوبة

من قبل هذا التصميم تعوّض أحداث المدوديّة والصيانة للخطر». وبالفعل فإن دعم تطبيقات Windows و MS-DOS و OS/2 إضافة إلى POSIX بإستعمال هذه البنية كان أمراً مستحيلاً.

لذلك عمد المصمّمون إلى إكتشاف بدائل أخرى. لقد كانوا بحاجة لطريقة تفصل الآليات المرجعيّة لإدارة المعالجة والذاكرة ومناولة الإستثناء وما شابه من المبادىء التي تتحكّم بكيفيّة عرض هذه الآليات إلى البرامج التطبيقيّة. إضافة لذلك، إحتاج المصمّمون لطريقة تفصل البيانات العامة والمبادىء المطلوبة من قبل طبقات API المختلفة عن NT لإبقاء النظام NT صغيراً أو غير فوضوي.

لقد حلّ نظام التشغيل Mach بنجاح بعض هذه المشاكل في الثمانينات وهو إستخدام مستضاف / ملقّم للنظام UNIX الذي يفصل آليات نظام التشغيل مثل إدارة الذاكرة وجدولة الشعَب من مختلف روتينات API في UNIX (وغير العائدة إلى UNIX) التي توفّرها الملقّمات. إستعار مصمّموا Windows NT طريقة Mach واستعملوها. يظهر التصميم الناتج، مع الإضافة الحَرِجة للنظام الفرعي 32-bit Windows، في الشكل (4-5).



الشكل (4-5)
تصميم النظام الفرعي المحمي في Windows NT

إن البرنامج التنفيذي NT هو نظام تشغيل شامل عام الأغراض. وهو يزوّد آليات نظام التشغيل المرجعيّة ويتيح للأنظمة الفرعيّة للمحيط إستعمال الألسنيّة والمبادىء للتطبيقات التي تدعمها. إن كل نظام فرعي لمحيط هو نظير للأنظمة الفرعيّة الأخرى ويستطيع إستدعاء خدمات NT المحليّة التي يحتاجها لإنشاء محيطات تنفيذ مناسبة لتطبيقاته المستضافة.

2-1-1-5 حماية الذاكرة:

أدّى نقل روتينات API إلى خارج البرنامج التنفيذي NT إلى فصل جيّد بين ترتيب نظام التشغيل والألسنيّة والمبادىء المطلوبة من قبَل روتينات API المختلفة. والفائدة الأخرى لهذه البنية هي أنها سهّلت هدفاً مهماً آخر للنظام Windows NT وحماية كل محيط API من تطبيقات المستعمل وحماية البرنامج التنفيذي NT من المحيطات.

يستعمل نظامَي OS/2 و 16-bit Windows نموذج DLL لاستخدام روتينات API. وفي ذلك النموذج، يتوفّر روتين API في DLL واحدة أو أكثر مشاركة والتي ترتبط بها البرامج التطبيقيّة وتستدعيها كإستدعاءات إجراء عادية. يعدّل النظام الصورة المنفّذة للمستدعي للإشارة إلى مقاطع DLL المشاركة خلال وقت التشغيل. (راجع الشكل (5-5)).

رغم أن نظامي OS/2 و Windows يستعملان DLL بطريقة مختلفة، غير أن النتيجة هي نفسها. فكل تطبيق يرتبط مع DLL يستطيع تعديل البيانات المستعملة من قبَل كل التطبيقات. وفي Windows NT، فهذا الوضع غير مقبول لأن إثنين من المتطلّبات المهمة هما القوة والأمان. فلا يجب على برنامج تطبيقي أن يؤثّر سلباً على نظام التشغيل أو على أي تطبيق آخر. فمثلاً، تتعقّب شيفرة Windows بعدد الأُطر على الشاشة. فإذا أراد برنامج مستعمل الكتابة فوق هذه البيانات، قد يعلّق النظام Windows أو يصبح غير منتظم، الأمر الذي يؤدي إلى إيقاف التطبيقات أو مقاطعة تنفيذها. إن النظام Windows NT لا يجتنب الروابط DLL، لكن طريقة إستعمالها في OS/2 و Windows تظهر مشكلة حماية الشيفرة والبيانات في نظام التشغيل. وليكون



الشكل (5-5)
نموذج DLL لروتينات API

نظام تشغيل أمن كلياً، يجب منع الوصول إلى شيفرة نظام التشغيل وبياناته على تطبيقات نمط المستعمل.

إن الحل لمشكلة حماية الذاكرة هي نموذج المستضاف / الملقّم للنظام Windows NT. فكل نظام فرعي محمي يشتغل في معالجة مع فسحة عنوان خاص. وليتمكّن تطبيق من الوصول إلى نظام فرعي، عليه أن يرسل رسالة. يستلم الملقّم هذه الرسالة، ويحدّد صلاحية كل البارامترات وينفّذ الوظيفة المطلوبة ويرجع النتائج إلى المستدعي. وبإستعمال هذا الإجراء، لا يتوفّر للمستدعي وصول مباشر إلى فسحة عنوان النظام الفرعي. ويستطيع فقط النظام الفرعي الوصول إلى البيانات العامة التي يحافظ عليها.

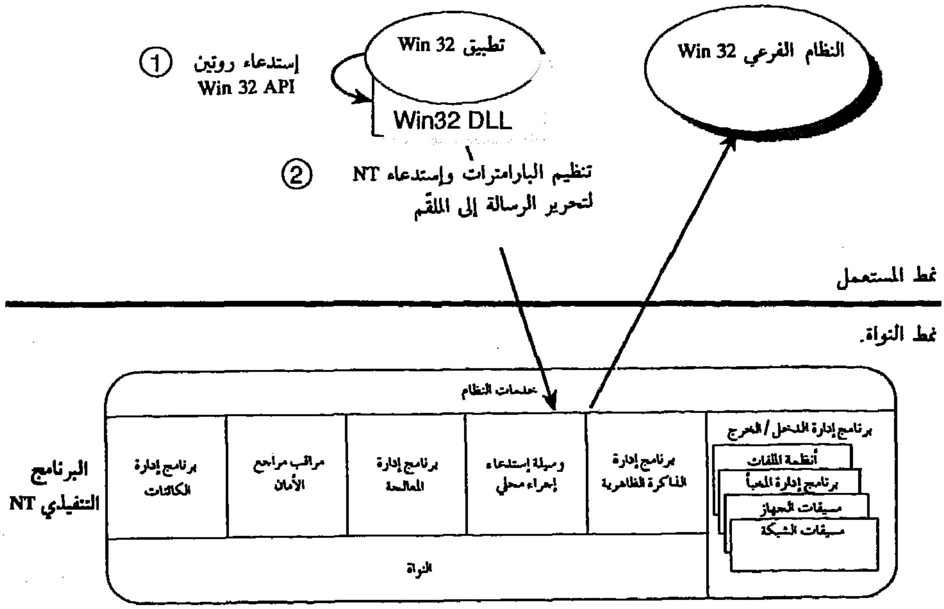
للبرهة الأولى، قد يبدو أن التطبيق الذي يشتغل على Windows NT يحتاج لإعادة كتابة لتمرير الرسائل إلى الملقّمات عوضاً عن إستدعاء روتينات API، لكن ذلك ليس صحيحاً. فالتطبيقات ما زالت تترابط مع DLL كما في السابق، وكل DLL يحتوي نقاط إدخال API تسمى جذول تنظم بارامترات المستدعي في رسالة وترسل الرسالة إلى الملقّم الصحيح. يستخدم الملقّم روتين API ثم يرجع النتائج إلى شيفرة DLL عبر البرنامج الخدماتي LPC. يرجع DLL بالطريقة العادية إلى التطبيق بحيث يكون تمرير الرسائل مرئياً لمبرمج التطبيق. يوضح الشكل (6-5) هذا التصرُّف مع النظام الفرعي Win 32. تعمل الآلية بنفس الطريقة للأنظمة الفرعيّة المحميّة الأخرى.

بإستعمال هذا النموذج، يتعذّر على تطبيق Win 32، على سبيل المثال، تشويه البيانات العامة للنظام الفرعي Win 32 والتأثير سلبياً على التطبيقات الأخرى. إضافة لذلك، يفصل كل نظام فرعي وبالتالي يحمى من الأنظمة الفرعيّة الأخرى. ويستطيع كل نظام فرعي إنشاء وصيانة بنيات البيانات بشكل مستقلّ وتحقيق أي ألسنيّة خاصة تحتاج لها بنية المعالجة ومناولة الإستثناء والخطأ والدخل / والخرج وما شابه.

إضافة لذلك، ولأن النظام الفرعي هو تطبيقات في نمط المستعمل، فإنها لا تستطيع تعديل بنيات البيانات للبرنامج التنفيذي أو إستدعاء روتينات نظام تشغيل داخليّة. والطريقة الوحيدة التي تستطيع الوصول عبرها إلى البرنامج التنفيذي NT هي عن طريق إستدعاء خدمات النظام. ولا حاجة لبنية حلقيّة معقّدة أو آليّة وقاية أخرى. يفصل نموذج المستضاف / الملقّم بين الأنظمة الفرعيّة وقسم نمط النواة من النظام.

الفائدة الأخيرة لإستعمال نموذج المستضاف / الملقّم هي أنه يمكن لأي عدد من الأنظمة الفرعيّة الإشتغال في نفس الوقت، بحيث توفّر محيطات API متعدّدة. كل نظام فرعي هو معالجة

في نمط المستعمل بشعَب يمكن جدولتها بشكل مستقل عن معالج، بحيث تحسّن التوازي في النظام. وبكل هذه الفوائد، فإن نموذج المستضاف / الملقّم هو التصميم الأفضل للنـظام Windows NT.



الشكل (6-5)

نموذج DLL لروتينات API في Windows NT

2-1-5 إعتبارات الأداء:

لقد كان الأداء، الناحية المقلقة عند المطوّرين عندما تمّ إختيار نموذج المستضاف / الملقّم للنظام Windows NT. يتطلّب إستدعاء روتين API أو خدمة نظام على نظام تشغيل تقليدي عمليات أقل مما يتطلّبه إستدعاء روتين API في تشكيل مستضاف / ملقّم.

تستخدم أنظمة التشغيل الأحاديّة الطبقة والمتعدّدة الطبقات خدمات نظامها في قسم نمط النواة في النظام. فعلى هذه الأنظمة، وعندما تقوم شعبة في نمط المستعمل بإستدعاء خدمة، يحتجز العتاد الشعبة ويغيّر نمط المعالج إلى التنفيذ في نمط النواة. بعد ذلك ينفّذ نظام التشغيل الخدمة. وعند إنتهاء الخدمة، يحوّل نظام التشغيل المعالج إلى نمط المستعمل وتعاود الشعبة التنفيذ في شيفرة التطبيق. وفي معظم المعالجات، فإن هذا التتابع سريع جداً.

لكن في النظام Windows NT، وعندما يستدعي تطبيق Win 32 روتين Win 32 API، لا يستعمل روتين API في قسم نمط النواة من نظام التشغيل. وإذا راجعت الشكل (6-5)، ستلاحظ أن Win 32 DLL يستدعي خدمة نظام NT لإرسال رسالة. ترسل الخدمة رسالة إلى الملقّم ثم تنتظر أن يستلمها الملقّم، وينفّذ الخدمة ثم الإجابة. ولكي يحصل الملقّم على الرسالة وينفّذها، يجب أن يحصل تحويل سياقي ـ أي، يجب على البرنامج التنفيذي NT تنفيذ التتابع التالي:

1 ــ حفظ سياق شعبة المستضاف (حالة ماكنة متطايرة).
2 ــ إنتقاء شعبة ملقّم للتنفيذ وتحميل سياق شعبة الملقّم.
3 ــ تنفيذ روتين Win 32 API باستعمال شعبة الملقّم.
4 ــ حفظ سياق شعبة الملقّم.
5 ــ إعادة تحميل سياق شعبة المستضاف ومعالجة نتائج روتين API.

بالإعتماد على العتاد حيث نظام التشغيل مركّب، يضيف التحويل السياقي عمليات معالجة إضافيّة إلى مصيدة النظام. وبذلك، فإن الشعبة التي تستدعي روتين API مستعمل في ملقّم سيحقّق نجاحاً في الأداء في كل مرة يستدعي الروتين مقارنة مع إستدعاء روتين API مستخدم كمصيدة نظام. ولأن الأداء هو الأكثر أهميّة لنجاح النظام Windows NT، أمعن المصمّمون في دراسة هذا الموضوع قبل المتابعة إلى المواضيع الأخرى.

إن التحويل السياقي، أي عملية حفظ حالة الماكنة لشعبة واحدة وتحميل أخرى، هي عملية ثابتة الكلفة نسبياً. ووفقاً للمعالج، يمكن تنفيذ بعض الإستمثالات، مثل طلب عمليات التحميل والتخزين بذكاء وتحميل وتخزين أقسام سياق الشعبة المطلوبة فقط.

البرنامج الخدماتي لتمرير الرسائل هو متغيّر الأداء الآخر في إستدعاء من مستضاف إلى ملقّم والعكس. لقد أنشأ Steve Wood الذي صمّم البرنامج الخدماتي LPC واستخدمه، إنشاءه مع خيارات مرنة لإرسال البيانات. فمثلاً، يوفّر البرنامج الخدماتي LPC طريقة لإرسال رسائل قصيرة بسهولة وطريقة لإرسال الرسائل الطويل بشكل فعّال. لقد أنشئت الطريقة الثالثة لتمرير الرسائل خصيصاً لإستمثال الأداء في النظام الفرعي Win 32، وهذا هدف مهمّ لأن هذا النظام الفرعي يعالج كل دخل المستعمل ويولّد كل الخرج التخطيطي على النظام Windows NT.

إضافة إلى البرنامج الخدماتي لتمرير الرسائل المرن المستمثل، أدخل مطوّرو النظام Windows NT بعض «السمات» التي تخفض عدد التفاعلات التي ينفّذها المستضاف مع الملقّم:

- بإستعمال روابط DLL لجهة المستضاف لإستعمال روتينات API التي لا تستعمل البيانات العامة أو تعدّلها.

■ تخزين بيانات نظام فرعي معيّنة في البرنامج التنفيذي أو تخبّئه بيانات النظام الفرعي في روابط DLL لجهة المستضاف.

■ دفع إستدعاءات API للمستضاف وإرسالها إلى الملقّم في رسالة واحدة.

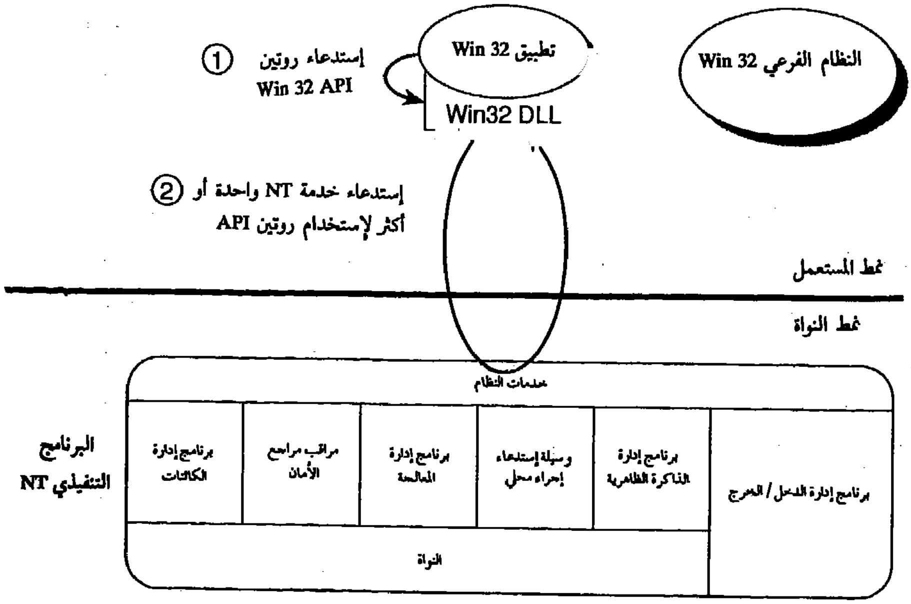
توفّر الخطة الأولى الفائدة الأكبر من بين الخطط الثلاث. وكما شرح سابقاً، إحدى أهداف إستعمال نموذج المستضاف / الملقّم هي لضمان عدم قيام المعالجات بتعديل البيانات العامة المحافظ عليها من قبل محيط معيّن. تشمل هذه البيانات معلومات مثل عدد الأُطر على الشاشة للنظام الفرعي Win 32 وجداول ترجمة المقبض للأنظمة الفرعيّة POSIX أو OS/2 والمعالجة الخاصة بالمحيط أو بطاقات تعريف الدورة التي تحافظ عليها كل الأنظمة الفرعيّة. ولأن هذه البيانات تستقرُّ في فسحة عنوان النظام الفرعي، يجب أن يمرّر المستضاف رسالة لتحقيق الوصول إلى هذه البيانات.

لكن عند التدقيق في روتينات API المتنوّعة، فإنه يلاحظ إمكانيّة تقسيم روتينات API إلى فئتين: الفئة التي تستعمل أو تغيّر البيانات العامة والفئة التي لا تفعل ذلك. ولا تحتاج روتينات API في المجموعة الأخيرة إلى إستدعاء النظام الفرعي. وهذا يعني أنه يمكن إستخدامها مباشرة في DLL خاص، كما يبيّن في الشكل (5-7). يستدعي DLL خدمات NT المحليّة للقيام بعملها بحيث تتجنّب التحويل السياقي وتمرير الرسائل.

لقد عمل مطوّرو الأنظمة الفرعيّة المتنوّعة في Windows NT بجهد لاستخدام روتينات API الأكثر إستعمالاً في DLL. وتستخدم روتينات Win 32 API التي تحصل على المعلومات المتعلّقة بتنفيذ الشعبة أو معالجتها، على سبيل المثال، أو حتى روتينات Win 32 API التي تضبط خصائص المعالجة أو الشعبة، وذلك في Win 32 DLL لأنها تستطيع إستدعاء خدمات NT للحصول على المعلومات التي تحتاجها دون الإنتقال إلى الملقّم Win 32، واعتمدت طريقة مشابهة للأنظمة الفرعيّة للمحيط الآخر وهي خطة تصنع أداء روتينات API الأكثر إستعمالاً ضمن نفس موقع تواجد إستدعاءات API على أنظمة تشغيل بطبقات متعدّدة أو أحاديّة الطبقة.

لقد أظهر التدقيق أن معظم روتينات API لا تتطلّب إستعمال البيانات العامة وبالتالي لا تتطلّب إستدعاء إلى الملقّم. أما روتينات API التي تتطلّبها فهي عمليات غير مستعملة كثيراً أو هي خدمات «عالية الكلفة» مثل إنشاء معالجة أو فتح ملفّ. ولا تذكر كلفة التحويل السياقي وتمرير الرسائل في هذه الحالات إلى المستعمل.

إن إستمثالَي المستضاف / الملقّم الثاني والثالث هما أصغر لكنهما ليسا أقلّ أهميّة. يخزّن الإستمثال الثاني بيانات النظام الفرعي في البرنامج التنفيذي أو يخبّئها في DLL. إن أحرف



الشكل (7-5)
إستخدام روتين API في DLL لجهة المستضاف

السواقة، كما ذكر في الفصل الثالث، «برنامج إدارة الكائنات وأمان الكائنات» هي مثال عن بيانات النظام الفرعي المخزّنة في البرنامج التنفيذي NT. يتطلّب كل من النظامين OS/2 و MS-DOS رموزاً عامة لأحرف السواقات A و B و C وأي أحرف أخرى يمكن أن ينشئها المستعمل، ولتجنّب إستدعاء الملقّم في كل مرة يشير تطبيق إلى سواقة، تنشأ أحرف السواقة ككائنات مسمّاة من قبَل NT والتي تُترجم أسماؤها من قبَل برنامج إدارة الكائنات إلى مقاصد الجهاز خلال وقت التشغيل. وبشكل مشابه، يخبىء النظام الفرعي Win 32 البيانات العامة في فسحة عنوان التطبيق. فمثلاً، عندما يسحب تطبيق كائن، يحتفظ النظام الفرعي Win 32 بنسخة عن الكائن في DLL لجهة المستضاف بحيث يحتوي DLL على البيانات المطلوبة لتنفيذ مزيد من روتينات API دون طلب معلومات من ملقّم Win 32.

يدفع الإستمثال الثالث، الذي صمّم من قبَل Chuck Whitmer، مهندس القسم التخطيطي في النظام الفرعي Win 32، المعلومات في المستضاف ويرسلها كرسالة واحدة إلى الملقّم. فمثلاً، قد يستدعي التطبيق عدّة روتينات رسم خط متتالية. يجمّع DLL هذه الروتينات في دفعة واحدة، ويرسل الإستدعاءات إلى الملقّم في رسالة واحدة. أو قد يضبط التطبيق لون القلم. ويتذكّر DLL أن لون القلم قد تغيّر حيث يرسل فقط تلك المعلومات إلى الملقّم عندما

يرسم في المرّة المقبلة شيئاً ما على الشاشة. يؤدي دفع إستدعاء أو معلومات API بهذه الطريقة إلى إرسالات أقل للبيانات الكبيرة بين المستضاف والملقّم، وبالتالي يحقّق أداءًا أفضل في عمليات الرسم.

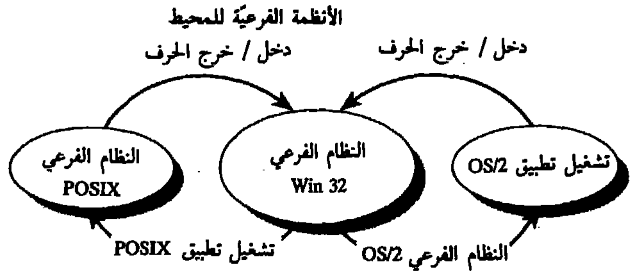
لقد كانت الحماية المحسنة والقوّة المضمونة باستعمال نموذج المستضاف / الملقّم في Windows NT مهمّة بحيث قبِلَ المطوِّرون تخفيضاً بنسبة 10% في سرعة تنفيذ التطبيق مقارنةً مع سرعة التطبيقات المشغّلة على Windows 3.1. لكنّ تحسينات الأداء المسردة هنا خفّضت إلى حدٍّ كبير عدد روتينات API التي تستدعي الملقّم فعلياً. وتلك التي تستدعيه تستمثل باستعمال طرق الدفع والتخبئة وشكل منسّق لإستدعاءات LPC، (المشروحة في القسم (5-5)). وباستعمال هذه الطرق (وأخرى)، تنفّذ مجموعة بسيطة من إستدعاءات API مجهوداً كبيراً وهذا المجهود يهبط إلى علامة 10%.

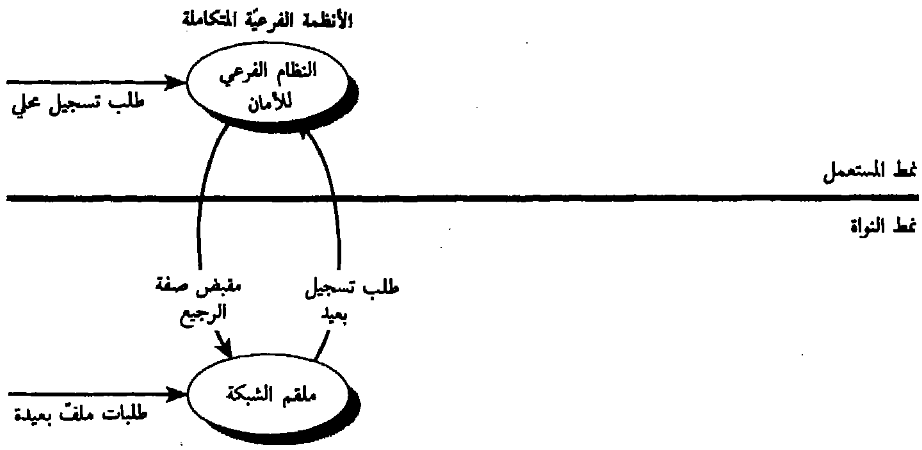
2-5 التفاعل مع الأنظمة الفرعيّة Windows:

إضافة إلى تفاعل الأنظمة الفرعيّة في Windows NT مع تطبيقات المستضاف عندما تستدعي التطبيقات روتينات API، تتفاعل الأنظمة الفرعيّة أيضاً مع بعضها البعض في طرق متوقّعة. يوضّح الشكل (5-8) بعض الأنظمة الفرعيّة المحميّة في Windows NT والتفاعلات النموذجيّة التي تحصل بينها.

إضافة للمسؤوليّات الأخرى، يتحكّم النظام الفرعي Win 32 بالتداخل مع المستعمل. وهو يدير الأُطر على الشاشة ويعرض الخرج للأنظمة الفرعيّة للمحيط الأخرى ويلتقط الدخل من المستعملين ويوجّهها إلى النظام الفرعي أو التطبيق الصحيح. يبدأ النظام الفرعي Win 32 التطبيق إستجابةً للطلبات من برنامج إدارة البرامج Windows Program Manager أو التطبيق الذي يحوي الأمر (الكونسول).

إضافة إلى الأنظمة الفرعيّة للمحيط المبينة في صفّ في أعلى الشكل، يُرسم نظامان فرعيان متكاملان. وهذه هي مكوّنات النظام التي تستفيد من الحماية المتوفّرة من قبل بنية مستضاف / ملقّم وقدرتها على الجدولة بشكل مستقلّ للتنفيذ. تعالج الأنظمة الفرعيّة للأمان تسجيلات المستعمل وتنشيء صفات أمان لعرض معالجات المستعمل وتحافظ على قاعدة بيانات معلومات الأمان المتعلّقة بحسابات المستعملين. تستجيب الأنظمة الفرعيّة للشبكة للطلبات التي تصل من الشبكة. يمكن تحميل أكثر من نظام فرعي لشبكة واحدة (يُعرف عادة بإسم ملقّم الشبكة) في Windows NT لمناولة الطلبات الصادرة عن الشبكات المختلفة.





الشكل (5-8)
تفاعلات الأنظمة الفرعيّة

تشرح الأقسام الفرعيّة التالية بتفصيل أكبر نوعين من التفاعل الذي يحصل ضمن الأنظمة الفرعيّة عندما يدخل المستعمل أنواع معيّنة من الدخل. ويحصل أول نوع تفاعل النظام الفرعي عندما يحاول مستعمل التسجيل إلى النظام Windows NT ويحصل نوع التفاعل الثاني عندما يشغّل مستعمل برامج تطبيقيّة غير برامج Win 32 التطبيقيّة.

1-2-5 التسجيل:

تسجيل المستعمل هي مزيّة جديدة في النظام Windows NT، صمّمت لجعل نظام التشغيل آمناً (وفقاً لمواصفات الحكومة الأميركيّة). يمنع التسجيل المستعملين غير المرخّص لهم من القيام بأشياء عرضيّة أو عن قصد لا يفترض أن يقوموا بها على بيانات مستعمل آمن.

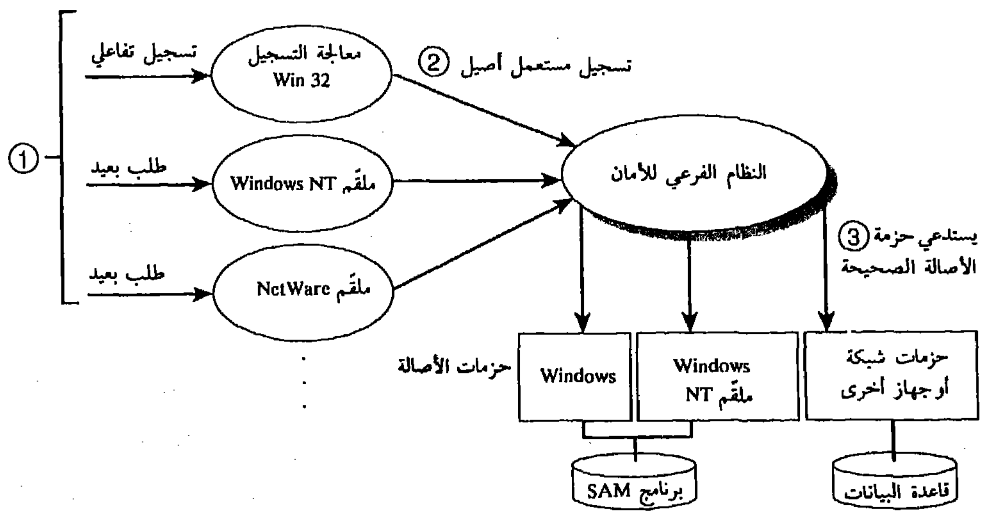
توجد طريقتان للوصول إلى Windows NT: عبر تسجيل تفاعلي أو التسجيل على شبكة.

يضمن النظام الفرعي للأمان حصول أي مستعمل يحاول الوصول إلى Windows NT على إذن للقيام بذلك من قبل مدير النظام. وفي معظم الحالات، فـ«الإذن» يعني وجود إدخال للمستعمل في برنامج إدارة الحساب للأمان في Windows NT (SAM)، وقاعدة بيانات تحتوي أسماء المستعملين وكلمات السرّ ومعلومات الأمان الأخرى.

صمَّمَ كلٌّ من Jim Kelly و Clif Van Dyke وهما مطوّران بخبرة عدّة سنوات في حقل الأمان وإنشاء الشبكات النظام الفرعي للأمان. وكان أحد الأهداف العديدة في تصميمهم هو ضمان مرونة النظام Windows NT في عدّة أنواع أجهزة التسجيل الخارجيّة التي يدعمها. ولتحقيق ذلك، إتخذ تصميم التسجيل الشكل المبيّن في الشكل (9-5) على الصفحة التالية.

يمكن إدخال طلبات المستعمل إما من لوحة مفاتيح مثبّتة بالنظام Windows NT أو على شبكة. تظهر طلبات الشبكة نفسها كطلبات للتوصيل إلى موارد شبكة و / أو لتنفيذ عمليات دخل / خرج. ولكل من طلبات التفاعل والشبكة، يجب أن تتوسّط معالجة محليّة وتتأكّد من شرعيّة الوصول. ينفّذ ملقّم Windows NT المركّب بالداخل أو ملقّم الشبكة الآخر هذه المهمّة لطلبات الشبكة.

وبالنسبة للطلبات التفعاليّة، تنتظر معالجة Win 32 قيام المستعمل بكبس توليفة المفاتيح Ctrl-Alt-Del. وعندما تكتشف هذا الإدخال، تحث معالجة التسجيل المستعمل لإدخال معلومات التسجيل.

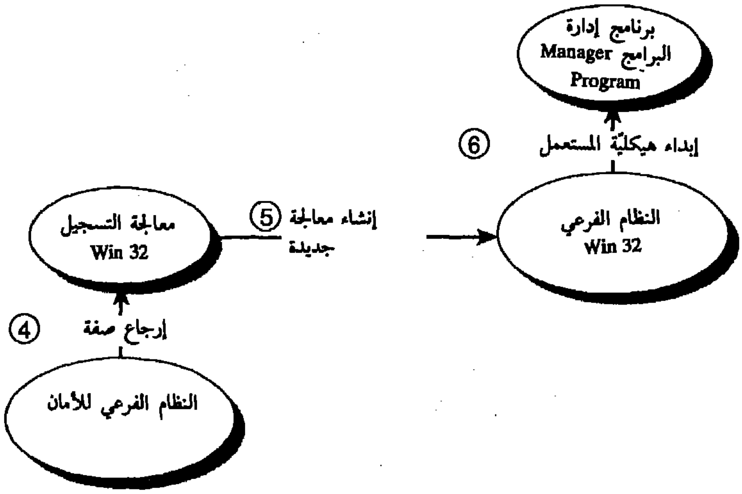


الشكل (9-5)
التسجيل والنظام الفرعي للأمان

يحتاج إلى نوعين من المعلومات للتحقّق من هوية المستعمل: معلومات التعريف ومعلومات الأصالة، الأولى هي إسم حساب المستعمل والثانية هي كلمة السرّ. لكن النظام الفرعي للأمان Windows NT مرنٌ بحيث يمكن أن تتّخذ معلومات التعريف شكل بطاقة ATM، مثلاً، ويمكن أن تكون معلومات الأصالة رقم معلومات شخصيّة. وبشكل مشابه، إذا استعمل ماسح لشبكة العين أولبصمة الإصبع لتعريف المستعملين، قد تكون معلومات بطاقة التعريف هي إسم المستعمل ومعلومات الأصالة التي يمكن أن تكون صورة العين أو بصمة الإبهام.

بعد أن يستلم النظام الفرعي للأمان معلومات التسجيل، فإنه يستعمل حزمة أصالة لتحقّق من المعلومات. يمكن قبس حزمات أصالة مختلفة في نظام الأمان في Windows NT بحيث تدعم أجهزة الدخل المستقبليّة بسهولة. لكن في التسجيلات العاديّة، يدقّق Windows أو حزمة الأصالة للشبكة بقاعدة بيانات SAM وإذا تطابقت كلمة السرّ المدخلة مع تلك الموجودة في قاعدة البيانات، ترجع حزمة الأصالة بطاقة تعريف المستعمل وقائمة ببطاقات تعريف المجموعة التي ينتمي إليها المستعمل.

بعد ذلك يحصل النظام الفرعي على معلومات إضافيّة حول المستعمل من قاعدة بيانات القوانين المحليّة بما فيها أية تفضيلات مخصّصة وحدود الحصص. وأخيراً، ينشىء النظام الفرعي للأمان صفة وصول تمثّل المستعمل والمسارات التي تتناولها الصفة إلى معالجة التسجيل، كما يبيّن في الشكل (5-10). وبذلك، يكون المستعمل أنشأ دورة تسجيل مع Windows NT.



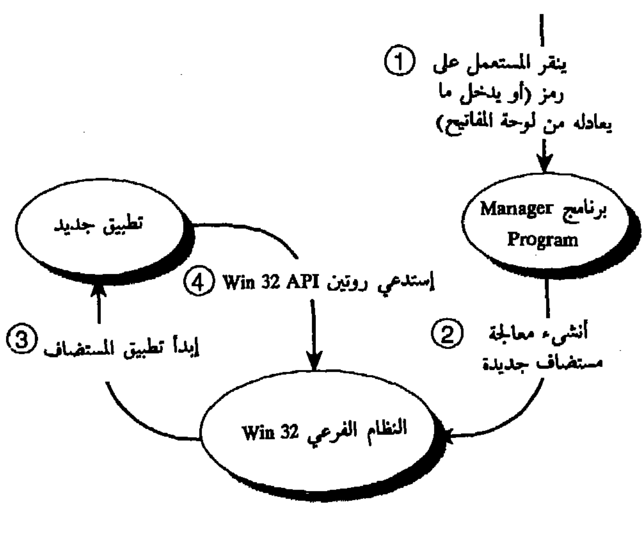
الشكل (5-10)
بدء دورة مستعمل تفاعلي

بالنسبة للتسجيلات التفاعليّة، تستدعي معالجة التسجيل Win 32 النظام الفرعي Win 32 لإنشاء معالجة جديدة وإلحاق صفة المستعمل بها. يبدأ النظام الفرعي هيكليّة مستعمل في المعالجة. وتكون هذه الهيكليّة عادة هي Program Manager، كما يظهر الشكل (5-10)، رغم أنه يمكن أن يكون POSIX أو نوعاً آخر من هيكليّة الأمر.

بالنسبة لتسجيلات الشبكة، يستعمل ملقّم الشبكة صفة الوصول لتقليد المستعمل والوصول إلى موارد النظام. ويستطيع بعد ذلك نسخ ملفّ أو تنفيذ أي عمل طلبه المستعمل البعيد.

5-2-2 تشغيل التطبيقات:

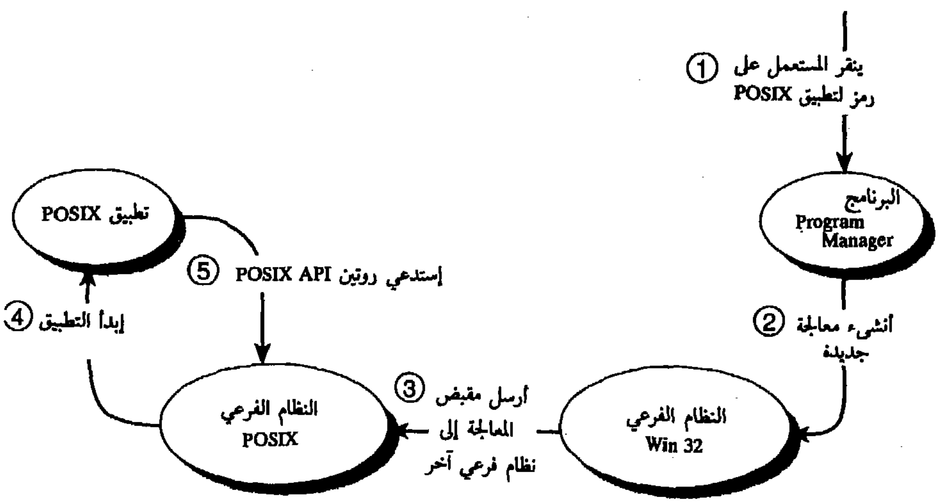
بعد تسجيل مستعمل تفاعلي، ينشىء النظام الفرعي Win 32 معالجات لتشغيل التطبيقات المختلفة التي يبدأها المستعمل. فعندما ينقر المستعمل على رمز، مثلاً، يوجّه النظام الفرعي Win 32 دخل الماوس إلى تطبيق Program Manager. كما يوضح الشكل (5-11) على الصفحة التالية، يستدعي Program Manager بدوره النظام الفرعي Win 32 لإنشاء الأُطر وإرسال الرسائل وما شابه. وعندما يدخل المستعمل دخلاً، يوجّه النظام الفرعي Win 32 الدخل إلى التطبيق الصحيح.



الشكل (5-11)
بدء تطبيق Win 32

إن النظام الفرعي Win 32 هو الرابط بين المستعمل وبقيّة نظام التشغيل. أما التطبيقات التي تستدعي روتين Win 32 API فهي مستضافات للنظام الفرعي Win 32 وهي «ملقّمة» مباشرة بواسطته. لكن النظام الفرعي Win 32 لا يستطيع تشغيل التطبيقات الأخرى مباشرة لأنه لا يستخدم روتينات API 16-bit Windows و MS-DOS و OS/2 و POSIX. لذلك، وعندما يبدأ مستعمل تطبيق ذات نسق ملفّ صوَر لا يتعرَّف إليه النظام الفرعي Win 32. ينشىء النظام الفرعي معالجة لتشغيل التطبيق، ولكن وعِوَضاً عن بدئه، يمرّر النظام الفرعي للتحكُّم بالمعالجة إلى نظام فرعي آخر، كما يظهر في الشكل (5-12). بعد ذلك، تنتقل إستدعاءات API العائدة للتطبيق مباشرة إلى النظام الفرعي الذي يستخدم روتينات API التي يطلبها التطبيق.

لا يتيح النظام Windows NT للتطبيقات إستدعاء روتينات API من محيطات أنظمة تشغيل مختلفة (مثلاً Win 32 و POSIX) لأن لا معنى لذلك من ناحية نقليّة التطبيق ولأنه لن يعمل بشكل صحيح. يحتفظ كل نظام فرعي بمفهومه حول الذي يؤلّف معالجة أو مقبض ملفّ، مثلاً، ولا يحتمل لبنيات البيانات المستعملة في محيط واحد أن تطابق تلك الموجودة في محيطات أخرى. لذلك، وبعد أن يعيّن نظام فرعي Win 32 معالجة تطبيق إلى نظام فرعي آخر، يبقى التطبيق مستضافاً لذلك النظام الفرعي إلى أن تنتهي المعالجة. يستمرُّ النظام الفرعي Win 32 بتوجيه دخل المستعمل إلى تطبيقات.



الشكل (5-12)
بدء تطبيق غير Windows

إضافة لإدارة دخل المستعمل، يعرض النظام الفرعي Win 32 التطبيقات كنوع من إثنين: تخطيطيّة أو حرفيّة.

إن تطبيقات Win 32 و 16-bit Windows هي تطبيقات تخطيطيّة. وهي تستعمل القوائم وخانات الحوار، وهي ترسم النص والخطوط في إطار وما شابه. لكن من الناحية الأخرى، فإن التطبيقات الحرفيّة تكتب خرج نصٍّ حرفيّ إلى الشاشة عند الوضعيّة الحاليّة للمشيرة. ويتمّ عادةً تشغيلها من مفسّر أمر وتخريج إلى مفسّر أمر عند الإنتهاء. إن تطبيقات MS-DOS مثل CHKDSK و FORMAT هي حرفيّة وكذلك هي تطبيقات OS/2 وكل تطبيقات POSIX (مواصفات IEEE 1003.1).

في النظام Windows NT، تحوّل التطبيقات الحرفيّة إلى تطبيقات تخطيطيّة لأن خرجها الخطي البسيط يعرض داخل إطار. ولتحقيق ذلك دون إعادة كتابة التطبيقات الحرفيّة، طوّرت Therese Stowell، مبرمجة سابقة في مجموعة أنظمة الملفّات OS/2، مجموعة من روتينات Win 32 API توجّه خرج التطبيقات الحرفية إلى أُطر نصيّة مُدارة من قبَل النظام الفرعي Win 32. تسمى هذه الأُطر كونسولات. فمثلاً، تستدعي مكتبة وقت التشغيل C روتينات كونسول لتوجيه الخرج القياسي لتطبيقات POSIX إلى إطار كونسول. وبشكل مشابه، عندما يستدعي تطبيق OS/2 وظائف VIO أو عندما يستدعي تطبيق MS-DOS وظائف INT(10)، يستدعي النظام الفرعي للمحيط OS/2 أو MS-DOS على التوالي روتينات كونسول Win 32 لعرض الخرج النصي.

تستقرُّ أُطر كونسول على جانب الأُطر التخطيطيّة في النظام Windows NT ويستطيع المستعمل تمرير النصّ بين الإثنتين عبر الحافظة Win 32 Clipboard، إضافة لذلك، ومع توفّر روتينات كونسول Win 32 الجديدة، يستطيع المطوِّرون كتابة تطبيقات حرفيّة من 32 بتاً للنظام Windows NT. إن معظم البرامج الخدماتيّة لخطّ الأوامر المزوّدة بـWindows NT هي تطبيقات Win 32.

3-5 النظام الفرعي win 32:

رغم أنه يشغّل عدّة أنواع مختلفة من التطبيقات، غير أن Windows NT هو أول نظام تشغيل Windows. وبالتحديد، فإنه نظام تشغيل Windows متطوّر في عائلة أنظمة Windows. وبإصدار Windows NT، تستطيع الحواسيب من المفكرات الصغيرة إلى الكبيرة وحواسيب محطة العمل المتعدّدة المعالجات، تشغيل كل تطبيقات Windows. وبالنسبة لمطوّري التطبيقات، فإن ذلك يعني أنه يُتاح لجهد تطوير تطبيق واحد الإشتغال على مجال واسع من عتاد الحواسيب.

كما سبق وذكر في هذا الكتاب، بدأ فريق Windows NT كتابة النظام على أنه سيكون نظام تشغيل OS/2 متطوّراً يدعم أيضاً POSIX API. لكن التحويل من OS/2 إلى Windows تمّ في منتصف طريق تطوير Windows NT، رغم أن التغيير بالنسبة للطاقم ولإدارة المشروع (أي، وجب على فريق OS/2 التخلّص من شيفرته وبدء شيفرة جديدة)، كان غير منطقي لناحية تصميم نظام تشغيل. فيمكن لنظام فرعي لمحيط Windows القبس في البرنامج التنفيذي NT وإستبدال النظام الفرعي OS/2. ولأن النظام الفرعي Windows سيكون نظام التداخل التخصيصي والبرمجي مع المستعمل كتطبيق أولي للنظام Windows NT، كان من الضروري توفير محيط «Windows فائق» يمدّد القدرات المتوفّرة في نظام 16-bit Windows NT مع روتينات 32-bit API والمزايا المتقدمة الأخرى.

لذلك، عند إنشاء محيط Windows جديد، بدأ Scott Ludwig و Chuck Whitmer والآخرون في فريق 32-bit Windows, Leif Pederson، معاينة براجيات Windows الموجودة عبر عدسات النظام Windows NT. فمحيط Windows على NT يجب أن يوفّر محيط برمجة مناسب لمحطات العمل المتطوّرة، وعليه أن يحقّق العديد من أهداف البرنامج التنفيذي NT، كالتالية:

- تحقيق نموذج ذاكرة خطيّة (مسطحة) من 32بتاً.
- إستخدام المهام المتعدّدة الشفعيّة.
- ضمان القدرة والأمان.

مع أن هذه الأهداف تمثّل بعض التغييرات على نظام 16-bit Windows، لكن بالنسبة لمطوّر أو مستعمل Windows، فإن النظام Windows NT هو محيط معروف.

يوفّر Windows NT قدرات إضافيّة لكنّه يحتفظ حيث أمكن بوظائفيّة Windows الموجودة. وبالنسبة للبرنامج التنفيذي NT، فالنظام الفرعي Win 32 هو تطبيق NT محلي معقّد. ولقد تمّ إعادة كتابته للنظام Windows NT ويستعمل خدمات NT محليّة كقاعدة له. رغم أن النظام الفرعي يستقرُّ في معالجة تطبيق، يعتمد Windows NT عليه للتفاعل مع المستعمل ولتوفير محيط برمجة وروتين API لتطبيقات أخرى.

يأخذ النظام الفرعي Win 32 إسمه من روتين 32-bit Windows API الجديد. يمدّد روتين API هذا، المتوفّر على Windows NT و MS-DOS و 16-bit Windows API ليس لكي يتمكّن من إستعمال نموذج الذاكرة المسطّحة 32-bit وحسب وإنما أيضاً ليزيد قدرات نظام التشغيل. يضيف روتين Win 32 API مزايا مثل الدخل / الخرج وإدارة الذاكرة المعقّدة وإدارة الكائنات والمعالجات المتعدّدة الشعَب والأمان وكذلك إدارة المخطّطات والنوافذ المحسّنة.

النظام الفرعي Win 32 ليس جديداً كلياً. فقد استعمل فريق المطوّرين شيفرة إدارة الأُطر وشيفرة التداخل مع المستعمل من Windows 3.0 واستعملوا أقصى ما يستطيعونه منه مع التخلّص وإعادة كتابة تلك الأجزاء التي يتعذّر جعلها قويّة وآمنة وشفعيّة وما شابه. كذلك إستخدم هذا الفريق مزايا Windows 3.1 بحيث يكون التداخل مع المستعمل في النظام Windows NT متوافق مع Windows 3.1. من الناحية المقابلة، فإن القسم التخطيطي في النظام الفرعي Win 32 هو كله جديد. فقد أعاد فريق مستقلّ تصميم محرّك المخططات من الصفر وصعوداً، حيث كتبوه بمعظمه باللغة C++.

توفّر الأقسام الفرعيّة التالية بعض المعلومات العامة حول Win 32 API حيث تحدّد البنية الأساسيّة للنظام الفرعي Win 32 وتشرح بتفصيل بعض الطرق التي يختلف فيها تصميم النظام الفرعي Win 32 عن تصميم النظام 16-bit Windows.

1-3-5 32-Bit API:

بذلت شركة Microsoft قدرة وموارد كبيرة لجعل Windows كمحيط تطوير للتطبيقات الشفعيّة. ففي العام 1990 بدأ نائب المدير الأول لقسم Steve Ballmer Microsoft Systems، إطلاق شعارات جديدة إلى كل من يسمع: «Windows, Windows, Windows» و «Windows في كل مكان». ورغم روحه الفكاهية التي أضحكت المجتمعين دائماً، غير أن شعاره الأخير ترك إنطباعاً مهماً. وكما ذكر الفصل الأول «المهمّة»، رأت Microsoft الحاجة لإنشاء نظام تشغيل متطوّر يستطيع إستخدام التقدّم العلمي في تقنيّة العتاد.

لقد كان روتين Windows 3.0 API الذي صُمِّم للإستعمال على أعلى MS-DOS، مقيّداً من هذه الناحية. ولكي يصبح محيط نظام تشغيل متطوّراً، كان من الضروري تطويره.

لقد كان من المفروض أن يوفّر Windows API محيط تطوير تطبيقات كامل ومعقّد، وغير محدود لثقافة البراامجيات القديمة أو يعتمد على أي تصميم معيّن للعتاد. ووجب عليه دعم كميات كبيرة من الذاكرة ومجموعة متنوّعة من المعالجات وحواسيب متعدّدة المعالجات، وإنشاء محيط آمن للتطبيقات.

رغم أن هذه الأهداف الطموحة، كان الهدف الأول لشركة Microsoft لتطوير Windows API ما يلي: جعل روتين API الجديد متوافقاً مع روتين 16-bit Windows API في أسماء الوظائف والألسنيّة وإستعمال أنواع البيانات حيث أمكن. ويجب على كل روتينات Win 32 API توفير مسار متحرّك صعوداً لنقل تطبيقات 16-bit Windows إلى Windows NT.

وباعتماد هذا الهدف، بدأ مطوّرو النظام Windows NT العمل لتحقيق أهداف إضافيّة لروتين Windows الجديد:

- تغيير روتين API ليستعمل بنية ذاكرة خطيّة من 32 بتاً. وعلى API قطع إعتماده على نموذج الذاكرة المقطعيّة المنتجة من قبل عائلة المعالجات Intel ∝86 لتحرير التطبيقات من تقييدات شيفرة 64 كيلوبايت وحدود البيانات والإتاحة لها تكبير نقليّتها وتوافقيّتها مع RISC ومنصّات العتاد اللا مقطعيّة الأخرى.
- جعل روتين 32-bit API هو نفسه على MS-DOS وعلى Windows NT بحيث يستطيع المطوّرون تشغيل تطبيقاتهم دون تعديل على مجال واسع من الحواسيب من النوع المادي إلى النوع المتطوّر.
- جعل محيط التطبيق آمناً باستخدام نظام الذاكرة الظاهريّة حيث يشغل كل تطبيق في فسحة عنوان خاصة به وبتوفير آليات حماية الكائنات في API.
- إضافة قدرات نظام تشغيل متطوّرة إلى API، مثل المعالجات المتعدّدة الشعّب وقدرات الدخل / الخرج المعتمد على API ومزامنة المعالجة وإدارة الذاكرة ودعم اللغة الوطنيّة (التدويل).

لإنشاء Win 32 API جديد، إستعمل مطوّروا شركة Microsoft ومديرو البرامج Windows 3.0 API وعدّلوه ليلائم الأهداف المُسردَة أعلاه.

بعد ذلك، إستخدمت شركة Microsoft حقول تجارب من داخل الشركة وخارجها للمساعدة في تحسين النتائج وتسهيل إستعمال المنافذ.

يوفّر النظام الفرعي Win 32 لكل التطبيقات على Windows NT. ولأن نظام الذاكرة الظاهريّة للبرنامج التنفيذي يعتمد على فسحة عنوان خطيّة لكل معالجة من 32 بتاً، تتعرّض التطبيقات التي تستدعي روتين Win 32 API لكلفة أقلّ من تلك التي تستعمل 16-bit Windows API أو MS-DOS API. لذلك، تشجّع شركة Microsoft المبرمجين على كتابة تطبيقات جديدة لتستعمل Win 32 API المتوفّر على Windows NT و MS-DOS.

تختلف وظائف Win 32 API في عدّة طرق متناسقة عن نفس وظائف API في Windows 3.0. أما الإختلافات الأكبر فهو توسيع بعض بنيات البيانات مثل المقابض والمؤشرات وإحداثيّات الرسم، من 16 بت إلى 32 بت وعدم إعتمادها على ذاكرة مقطعيّة. وفي الحالات حيث تؤثّر البارامترات الأوسع على التطبيقات الموجودة بحيث لا تنفصل

التطبيقات الموجودة وتتمكّن من الإنتقال إلى Win 32 API مع الوقت. فمثلاً، تمَّ تغيير البارامترات الصحيحة والمؤشّرات من مقطع: نسق جيّد إلى نسق مسطّح 32 بت والإحداثيات المستعملة في وظائف الرسم هي بعرض 32 بت بدلاً من 16 بت.

توفّر مجموعة جديدة من روتينات Win 32 API قدرات نظام تشغيل متقدّم، مثل الدخل / الخرج، والمزامنة وإدارة الذاكرة والأمان والشعَب. ورغم أنها مُصمَّمة لتشبه روتينات Windows API القديمة، فإن الخدمات الجديدة تصدر بشكل مباشر أو أقلّ مباشرة خدمات NT المحليّة، بحيث توفّر قدرة NT لمبرمجي Win 32.

ورغم إستعارة العديد من مزايا Win 32 هذه مباشرة من البرنامج التنفيذي NT، فإنه يُفاد إنشاؤها للنظام MS-DOS. تتوفّر مزايا معيّنة متطوّرة مثل قدرات الدخل / الخرج اللامتزامنة فقط على Windows NT.

توجد مزيّة واحدة جديدة يوفّرها Win 32 API ـ الأمان ـ في التداخل. إستخدمت مزيّة الأمان Win 32 من قبَل Jim Anderson وهو مطوّر له خلفيّة براجيّة في بناء حواسيب تتحكّم بالنوعيّة لمصانع محركات شركة Ford Motors Company. فمزايا الأمان التي طوّرها للنظام الفرعي Win 32 هي إمتدادات في نمط المستعمل إلى قدرات الأمان التي صمّمها Jim Kelly و Robert Reichel والآخرون في بنية كائنات البرنامج التنفيذي NT.

يستخدم النظام الفرعي Win 32 الأمان المعتمد على الكائن بنفس الطريقة التي يعتمدها البرنامج التنفيذي NT. فالنظام الفرعي Win 32 يحمي كائنات Windows المتشاركة من الوصول غير المسموح له بوضع واصف الأمان NT عليها. وكما في البرنامج التنفيذي NT، ففي أول مرة يحاول تطبيق الوصول إلى كائن مشارك، يتحقّق النظام الفرعي Win 32 من حق التطبيق للقيام بذلك. فإذا كان كذلك، يتيح النظام الفرعي Win 32 للتطبيق المتابعة.

يستخدم النظام الفرعي Win 32 أمان الكائن على عدد من الكائنات المشاركة، حيث البعض منها مركّب على أعلى كائنات NT المحليّة. تتضمّن كائنات Win 32، كائنات سطح المكتب وكائنات النافذة وكائنات القائمة ـ وكما في البرنامج التنفيذي NT ـ ملفّات ومعالجات وشعَب وعدّة كائنات مزامنة.

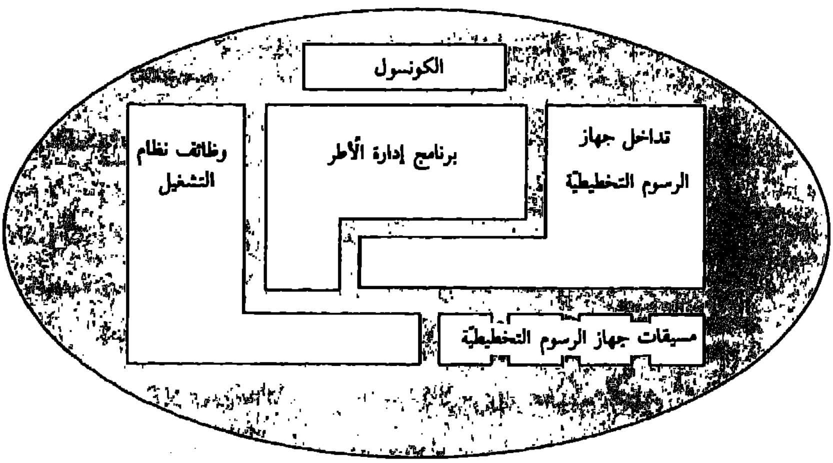
إن إعادة تصميم أجزاء معيّنة من التداخل التخطيطي مع المستعمل في Windows للنظام الفرعي Win 32 (وهو موضوع يوصَف لاحقاً) وإضافة مزايا أمان إلى Win 32 API، يجعل النظام الفرعي Win 32 والبرنامج التنفيذي NT قوياً وآمناً.

5-3-2 البنية:

يحتفظ النظام الفرعي Win 32 بالبنية الأساسيّة لنظام 16-bit Windows. وهو يتألف من المكوّنات المبيّنة في الشكل (5-13).

يقسم النظام الفرعي إلى خمسة أقسام منظوميّة: برنامج إدارة النوافذ الذي يتناول الدخل ويدير الشاشة، وتداخل الرسوم التخطيطيّة مع الجهاز (GDI) الذي هو مكتبة رسم لأجهزة خرج الرسوم التخطيطيّة، ووظائف نظام التشغيل، والكونسول الذي يوفّر دعم إطار النص، ومسيقات جهاز الرسوم التخطيطيّة Win 32. ويستخدم كل مكوّن روتينات API الذي يمكن أن يستعملها مبرمجو التطبيقات لإنشاء تطبيقات تخطيطيّة. وسوية، تؤلف تداخلات البرمجة هذه روتين Win 32 API.

إن برنامج إدارة الأُطر هو الذي يجعل النظام Windows NT يبدو مثل Windows. يتحكّم برنامج إدارة الأُطر بالأُطر على الشاشة، ويوجّه دخل المستعمل إلى التطبيقات وينشىء كائنات Windows قياسيّة ويرسل البيانات إلى الحافظة Clipboard ومنها يتناول المهام المرئيّة وغير المرئيّة. وهو يوفّر أيضاً روتينات API التي تسمح للتطبيقات إنشاء تداخلات تخطيطيّة مع المستعمل. فمثلاً، يمنع التطبيقات من التعامل مباشرة مع الأجهزة عن طريق توفير روتينات قياسيّة تستدعيها للحصول على معلومات من أجهزة الدخل. كذلك يقوم برنامج إدارة الأُطر بتعقّب الأُطر



الشكل (5-13)
النظام الفرعي Win 32

المعروضة على الشاشة وقياسها وطبقاتها. وعندما يكبّر المستعمل الإطار أو يصغّرها. يبلغ برنامج إدارة الأُطر التطبيق المتأثّر بذلك. وبشكل مشابه، يبلغ التطبيقات متى يجب إعادة طلاء الأُطر وعند قيامها بذلك، فإنها تعاود طلاء فقط أجزاء الأُطر التي يجب أن تكون مرئيّة. كذلك، يتيح برنامج إدارة النوافذ للتطبيقات قص المعلومات إلى الحافظة Clipboard ثم وضع المعلومات في دفق الدخل للتطبيق حيث يلصقها بالمستعمل.

يوفّر مكوّن GDI للنظام الفرعي Win 32 مجموعة غنيّة من روتينات API لرسم الخطوط والأشكال والرموز والنصّ على أجهزة خرج الرسوم التخطيطيّة (مثل العرض للفيديو أو الراسمة) ولتنفيذ المناولات المعقّدة للرسوم التخطيطيّة. يستدعي برنامج إدارة الإطار هذه الروتينات لرسم الأُطر والرموز الأخرى، ويستدعيها مكوّن الكونسول لرسم النص في إطار لكن التطبيقات تستطيع أيضاً إستدعاء روتينات GDI API مباشرة. ويستدعي GDI بدوره مسيقات جهاز الرسوم التخطيطيّة لعرض الأشكال والنصّ وتستدعي مسيّقات جهاز الرسوم التخطيطيّة مسيقات جهاز NT لمناولة عتاد الجهاز.

ومثل مكوّن الكونسول، فإن مكوّن نظام التشغيل Win 32 جديد بمعظمه. وهو يتيح لتطبيقات Win 32 تنفيذ دخل / خرج كامل المزايا ومناولة كائنات نظام التشغيل (إضافة إلى الكائنات التخطيطيّة) ومزامنة تنفيذ شعَبها مع أحداث النظام ومع التطبيقات الأخرى وإدارة الذاكرة بطريقة معقّدة ومشاركة الموارد بأمان وإنشاء تطبيقات متعدّدة الشعَب. تعتمد وظائف Win32 هذه على مزايا البرنامج التنفيذي NT وهي تستدعي خدمات النظام NT مباشرة.

3-3-5 تغييرات التصميم:

Windows/MS-DOS هو محيط نظام تشغيل منسّق صغير. وقد صُمِّمَ ليُشغَّل على حواسيب شخصيّة لا تحتوي على كميات كبيرة من الذاكرة ولا تحتوي المعالجات السريعة التي تنتجها شركتا INTEL وRISC حالياً. وعلى الحواسيب الصغيرة حيث كل بايت وكل دورة CPU مهمة، لم يكن عملياً جعل Windows محيط قوي بالكامل لأن القيام بذلك يؤدّي إلى كلفة معيّنة من ناحية قياس الشيفرة وسرعتها.

لكن النظام Windows NT، من الناحية الأخرى، مصمّم كنظام تشغيل يستطيع خدمة أي عدد من مُستَعمِلي الشبكات الذين يستطيعون تشغيل تطبيقات مصرفيّة أو حكوميّة حسّاسة خاصة. في مثل هذه المحيطات، ليس من المقبول الإتاحة لتطبيق واحد ليؤثّر على تطبيقات أخرى أو تعليق نظام التشغيل. لذلك كان جعل النظام الفرعي Win 32 قوياً هدفاً مهماً.

لحماية نظام التشغيل من التطبيقات، كان من الضروري تشغيل برنامج إدارة الأُطر في Win 32 بطريقة مختلفة قليلاً عن برنامج إدارة الأُطر الأصليّة. بدأ فريق Window Manager، بقيادة Scott Ludwig وهو عَمِلَ لثماني سنوات في برمجة Windows و Presentation Manager، في جعل إصدار 32-bit محيطاً أكثر قوّة واعتماديّة من برنامج إدارة 16-bit Window. وقد شملت التغييرات الأساسيّة ما يلي:

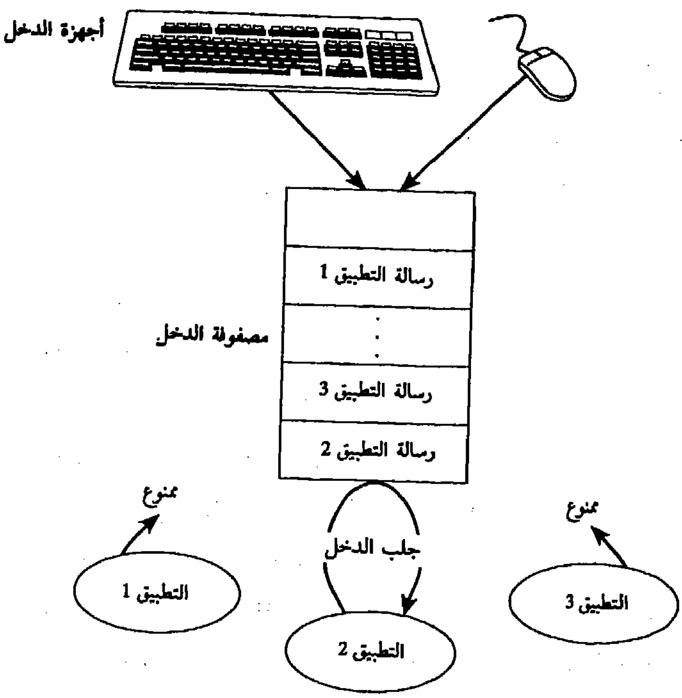
- إزالة مزامنة نموذج دخل Windows.
- تعيين جدولة شعَب شفعيّة.
- إضافة صلاحيّة مقبض الكائن وقفل الكائن.

يعود التغيير الأول إلى كيفيّة مناولة النظام الفرعي Win 32 لدخل المستعمل، مثل ضغوط المفاتيح ونقرات الماوس. يحتوي برنامج إدارة الأُطر 16 بت على نموذج دخل متزامن، وهذا يعني وضع كل دخل المستعمل في مصفوفة واحدة (الأولى الداخلة والأولى الخارجة) وإرسالها إلى التطبيقات عندما تتطلّبها. بعد إدخال الدخل، يجب أن ينتظر المستعمل إلى أن يعالجه التطبيق قبل معالجة أي إدخال لاحق. ولأنه يمكن لتطبيق واحد فقط في كل مرة إسترداد دخله، يستردّ كل تطبيق دخله من المصفوفة بطريقة زمنيّة لكي تنفّذ كل التطبيقات دون إعاقة. يوضّح الشكل (14-5) نموذج دخل المزامنة.

باستعمال هذا النموذج، حصلت أخطاء عديدة. مثل تشوُّش التطبيق وتوقُّفه عن إسترداد الدخل أو إنشغاله وعدم جلب إدخاله بسرعة. وعند حصول ذلك، قد تتوقّف تطبيقات أخرى لأنها لا تستطيع الحصول على الدخل المطلوب للمتابعة. وبالنسبة للمستعمل، يعلّق نظام التشغيل.

لقد تطلّب نموذج الدخل لبرنامج إدارة الأُطر 16 بت إعادة تصميم أساسيّة، وبالنتيجة تمّ تحسين قوة النظام الفرعي Win 32 إلى حدٍّ كبير. وفي نموذج الدخل الجديد، الذي صُمِّمَ من قبَل David Pehrson و Scott Ludwig، يحصل كل تطبيق على مصفوفة دخل خاصة، كما يبيِّن في الشكل (15-5) على الصفحة 151.

عندما يدخل مستعمل دخلاً، يحدّد فوراً النظام الفرعي Win 32 التطبيق الخاص بالدخل (عن طريق التدقيق بالإطار النشط حالياً أو بالإطار حيث مؤشّر الماوس موجود). يضع برنامج إدارة النوافذ الدخل في مصفوفة التطبيق الصحيحة ويستردّ التطبيق الدخل عندما يصبح جاهزاً. وإذا توقّف التطبيق عن إسترداد دخله لسببٍ ما، لا تتأثّر بذلك التطبيقات الأخرى.



الشكل (14-5)
نموذج دخل 16-bit Windows

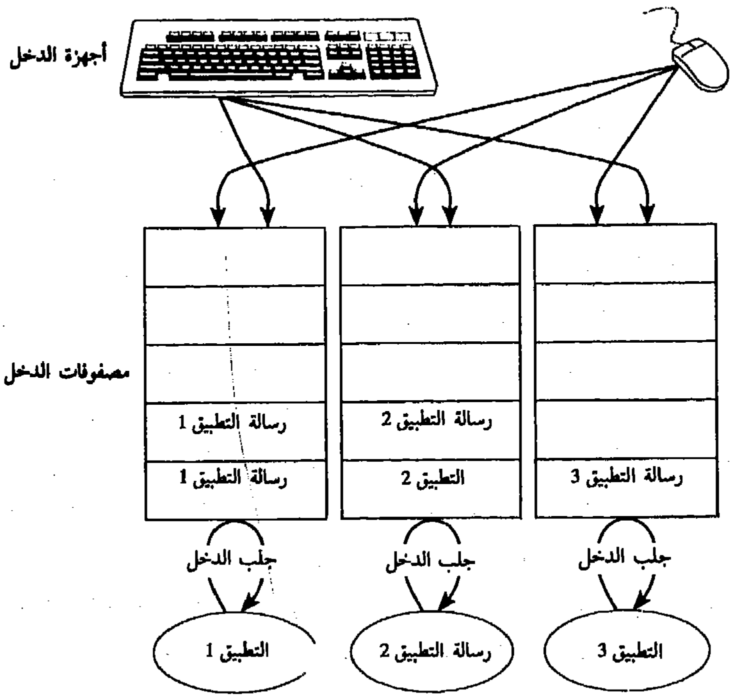
التغيير الرئيسي الثاني للنظام الفرعي Win 32 من 16-bit Windows هو المهام المتعدّدة بّة الكاملة. ففي نظام تشغيل قوي، ليس مقبولاً لتطبيق واحد التوقّف عن العمل مع عدم نظام التشغيل من دخوله أو إنهاء تنفيذ التطبيق. يعتمد النظام 16-bit Windows على قات ليتنازل عن المعالج أحياناً لكي تستطيع التطبيقات الأخرى من الإشتغال، وهو أمر بل دائماً. وفي النظام Windows NT يدفع النظام الفرعي Win 32 (المدعوم في النواة NT) ق ليتنازل عن المعالج. يُتاح لكل شعبة تطبيق العمل فقط لفترة كميّة الوقت الخاصة بها. لك تقاطع النواة NT الشعبة وتدقّق لجهة تشغيل شعبة بأولويّة أعلى.

ومع الشفعيّة الفعليّة، وإضافة إلى قيام النظام الفرعي Win 32 بدفع كل تطبيق ليتنازل لعالج، فإنه أيضاً يدفع تطبيق Win 32 للإنتهاء. فمثلاً، إذا علّق تطبيق أو إذا لم يعد يريده مل، يستطيع النقر على الزرّ End Task في Task Manager في Windows. وعند التشغيل النظام 16-bit Windows يمنع تطبيق الشيء نفسه من الإنتهاء أو في الحالات الأسوأ، يمكنه Task Manager من الظهور في المقدمة. لكن في النظام Windows NT، يستدعي النظام

الفرعي Win 32 البرنامج التنفيذي NT التي تعتمد على كائن شعبة NT، ويرسل برنامج إدارة المعالجة NT رسالة إلى منفّذ الإنهاء المسجّل للشعبة. يستلم النظام الفرعي للمحيط المسؤول عن التطبيق الرسالة وينظّف المعلومات العامة التي يحتفظ بها حول التطبيق المنهى.

إضافة إلى إزالة مزامنة نموذج الدخل وإستخدام الشفعية، يثبت النظام الفرعي Win 32 مقابض الكائن، وقد تسبّب مقابض النظام مشكلة أحياناً في النظام 16-bit Windows لأنه يفترض أن التطبيق يمرّر دائماً المقابض الصالحة ـ أي المقابض التي تشير فعلياً إلى الكائنات التي يتوقّع Windows أن تشير إليها. وإذا توقّع Windows مقبضاً إلى إطار ومرّره تطبيق كشيء لا يشير إلى إطار، قد تصبح براجميات Windows مشوّشة ومرتبكة.

لكن من الناحية المقابلة، فإن النظام الفرعي Win 32 يثبّت المقابض التي تمرّر إليها التطبيقات. وهي تقوم بذلك عن طريق التدقيق بمحتويات المقابض التي تستلمها.



الشكل (5-15)
نموذج دخل Win 32

ومثل مقبض كائن NT، فمقبض كائن Win 32 هو فهرس في جدول. يحتوي المقبض وإدخال الجدول الموافق له على معلومات خاصة بالكائن الذي يشير إليه المقبض. يستطيع النظام الفرعي Win 32 التحقّق من إشارة المقبض إلى كائن من النوع الذي يتوقّعه النظام الفرعي Win 32، ويستطيع أيضاً تحديد جهة إشارة المقبض إلى الحالة الآنيّة الصحيحة للنوع. إضافة إلى تثبيت صلاحية مقابض الكائن، يستخدم النظام الفرعي Win 32 شكل إحتجاز كائن مشابه لذلك في البرنامج التنفيذي NT. وفي النظام 16-bit Windows NT، يتعذّر تطبيق حذف كائن خلال إستعمال نظام التشغيل وهي حالة قد تؤدّي إلى أخطاء نظام مزعجة. لكن النظام الفرعي Win 32 يحافظ على عدّ مرجعي للكائنات المطلوبة. ويستطيع التطبيق حذف الكائن لكن النظام الفرعي لن يزيل الكائن من الذاكرة إلى أن ينتهي النظام من إستعماله.

لم يكن مكوّن Win 32 GDI تعديلاً لإصدار 16 بت ـــ وقد تمّ إعادة كتابته كلياً. صمّم الإصدار الجديد من قبَل Chuck Whitmer وتمّ كتابته من قبَل فريق مطوّرين بإدارة Kent Diamond. ولقد كانت الأهداف الأوليّة لمكوّن GDI للنظام الفرعي Win 32 هي في إستبدال شيفرة لغة assembly بشيفرة C النقّالة لإعادة تصميم العمل الداخلي GDI لدعم بعض القدرات المتقدّمة. تشغل مزايا GDI الجديدة منحنيات Bezier التي توفّر للمستعمل تحكُّماً دقيقاً في رسم الأقواس والممرّات التي تتيح للمستعملين إنشاء كائنات بأشكال إعتباطيّة باستعمال تتابعات أوامر الرسم. تحويل الكائن الذي يتيح للتطبيقات تخطيط محتويات فسحة إحداثيات واحدة في أخرى، ولاحقاً الترابط وهي طريقة للتطبيقات تجدّد بسهولة تراكب كائن واحد أو منطقة على بعضها البعض.

التداخل مع مسيق الجهاز الجديد هو تغيير التصميم الآخر في Win 32 GDI، الذي حقّق نتيجة الخبرة الجماعيّة لفريق GDI بمحركات الرسوم التخطيطيّة في 16-bit Windows و Presentation Manager. يحسّن هذا التداخل الجديد المحرّكين السابقين، حيث يوفّر لمسيقات جهاز الرسوم التخطيطيّة Win 32 (المسؤوليّة عن إنشاء رسوم خاصة بالجهاز وإرسالها إلى أجهزة الخرج) درجة أدقّ من التحكُّم. فمثلاً، يكيّف GDI الدخل الذي يوفّره إلى مسيق وفقاً لما يستدعيه المسيق. فمثلاً، إذا كان المسيق يستوعب منحنيات Bezier، يمرّر GDI منحنيات GDI الكاملة إليه كدخل. وإذا لم يكن المسيق يستوعب Bezier يقطع GDI المنحنى إلى مقاطع خطيّة صغيرة قبل إرسالها إلى المسيق. إضافة لذلك، يشمل GDI وظائف جديدة قوية لإنشاء رسوم نقطية التي يمكن أن يستعملها مسيقات الفيديو والطابعة بشكل خاص في كتابة وظائفها الخاصة. ومع هذا الدعم، يصبح تحفيز المسيقات وتشغيلها بسرعة أمراً سهلاً للمطوّرين. ويستطيع المطوّرون الإعتماد على GDI للقيام بمعظم العمل مع إضافة تحسينات خاصة على الجهاز والإستمثالات على مسيق الجهاز فقط.

رغم كل التحسينات التي ذكرت إلى الآن، لربما أكبر تغيير في برنامج إدارة النوافذ وفي GDI كان إستخدام Win 32 كنظام فرعي محمي، أي، معالجة ملقّم. وكما سبق وشرح في هذا الفصل، قسّم مصمّمو كل من الأنظمة الفرعيّة للمحيط في Windows NT روتينات API إلى مجموعتين: الروتينات التي تستعمل بيانات خاصة وتلك التي تستعمل بيانات عامة. ويمكن إستخدام المجموعة الفرعيّة السابقة في مكتبة DLL لجهة المستضاف لإستمثال الأداء. لكن يجب إستخدام المجموعة الفرعيّة الأخيرة في فسحة عنوان النظام الفرعي بحيث تتمُّ حماية البيانات العامة وتبقى متوفّرة لكل مستضافات النظام الفرعي. يؤثّر هذا التغيير في التصميم على كيفيّة إختيار كتابة تطبيق Win 32 وخاصة عند إستدعاء وظائف GDI.

مع أن GDI يعتبر ظاهرياً أن كل البيانات هي خاصة بمعالجة مستضاف، فعند رسم الكائنات، تشارك كل المعالجات جهاز الخرج ــ الشاشة. عندئذٍ تكون الشاشة «بيانات عامة» ويجب إستخدام وظائف GDI التي تغيّر حالة الشاشة في ملقّم Win 32 عوضاً عن مكتبة DLL لجهة المستضاف. ولتحقيق الأداء الأقصى لهذه الوظائف، يستخدم مكوّن GDI عدّة إستمثالات. فمثلاً، لتغيير الألوان على الشاشة، قد يستدعي تطبيق Win 32 وظيفة GDI لضبط لون الواجهة، وقد يستدعي مجدّداً لضبط لون الخلفيّة، ثم ينفّذ عدّة عمليات أخرى قبل رسم أي شيء على الشاشة. وعوضاً عن إستدعاء النظام الفرعي مرة واحدة لكل من وظائف GDI هذه، تخزّن مكتبة DLL لجهة المستضاف المعلومات المغيّرة في مخزن مؤقّت. وعندما يرسم التطبيق شيئاً ما على الشاشة، ترسل DLL كل البيانات المغيّرة إلى النظام الفرعي في رسالة واحدة. في المثال الحالي، يحدّث النظام الفرعي ألوان الشاشة في DLL، حيث ينتظر إرسال تغييرات الألوان إلى الملقّم إلى أن يرسم أول خطّ. تخبئة الصفة، كما تسمى طريقة التخزين المؤقّت هذه، تخفّض عدد التحويلات السياقيّة والوقت المستغرق لتمرير الرسائل بين المستضاف والنظام الفرعي.

يستعمل GDI إستمثالاً مشابهاً يسمى الدفع، لتخفيض التحويل السياقي بين المستضاف والملقّم. إن الدفع هي طريقة حيث تخزّن GDI DLL، على سبيل المثال، إستدعاءات الوظيفة المتعدّدة في مصفوفة، وترسلها إلى الملقّم في رسالة واحدة عندما تمتلىء المصفوفة أو عندما يدخل المستعمل دخلاً. وعندما يستلم ملقّم Win 32 الرسالة، فإنّه ينفّذ الوظائف في تتابع قبل إرجاع التحكّم إلى المستضاف. وقبل إستخدام هذه الطريقة، قام مطوّرو GDI باختباره للتأكّد من جودة خرج الشاشة. وقد أشار الإختبار إلى أن الخرج سلس ومُقاس.

يجب أن يتذكّر كاتبو التطبيقات إستمثالات الأداء هذه عند كتابة تطبيقات Win 32 متعدّدة الشعَب الجديدة. وإذا كانت شعبتان تعملان سويّة، يجب أن يزامنا تنفيذهما للتأكّد من تنفيذ

عملياتها بالترتيب الصحيح. فمثلاً، يجب أن يدرك أنه لأن شعبة إستدعت وظيفة API، فإن ذلك لا يعني أن النتيجة ستكون مرئيّة على الشاشة فوراً. يمكن أن يخزّن إستدعاء الوظيفة في المخزن المؤقّت للشعبة، بانتظار إرساله إلى الشاشه. يوفّر GDI الوظيفة () GdiFlush ليجعل الشعَب تدفع إرسال إستدعاءات الوظيفة المخبّأة إلى ملقّم Win 32.

التأثير الآخر لنموذج المستضاف / الملقّم هو أن التطبيقات التي تتناول بتكرار الرسوم النقطيّة الكبيرة وتُعاود رسمها، لا تعمل بشكل جيد مقارنة مع الأشكال المستقلّة عن الجهاز المرسومة باستعمال إستدعاءات GDI. ولأن كل كائن أو رسم نقطي هـو خاص لتـطبيق المستضاف، فإنه يجب أن يرسل رسالة إلى الملقّم في كل مرة يتمّ تحديث الشاشة فيها. وللمحافظة على الأداء الأمثل، يجب إما أن تقوم التطبيقات بإعادة رسم فقط الأقسام المعدّلة من رسم نقطي، إذا أمكن، أو الإعتماد على وظائف GDI لرسم كل الصوَر، مستفيدة من طرُق التخبئة والدفع في GDI لإستمثال الأداء. يجهّز روتين Win 32 GDI API الجديد بشكل أفضل في 16-bit Windows GDI API لتوفير كل قدرات الرسم التي تحتاجها التطبيقات المعقّدة.

4-5 MS-DOSو 16-bit Windows API:

قد يتساءل البعض «أن هذه المزايا المتطوّرة جيّدة ولكن هل يستطيع النظام Windows NT تشغيل تطبيقات MS-DOS و Windows المفضّلة؟». أنه سؤال مهمّ نسبة إلى التكاليف التي دفعها المستعملون لشراء التطبيقات الموجودة. من الواضح أن أكثريّة مستعملي Windows NT يعتمدون على تطبيقات MS-DOS و 16-bit Windows وسيعتمدون عليها لفترة طويلة قادمة. إن دعم هؤلاء المستعملين كان إعتباراً مهماً في تطوير Windows NT.

لحسن الحظّ، يستطيع نموذج المستضاف / الملقّم في Windows NT إستيعاب محيطات تنفيذ تطبيقات متعدّدة بسهولة. يتضمّن شمل محيط MS-DOS ومحيط 16-bit Windows، عملاً معقّداً ومفصّلاً لكنّه لا يغيّر تصميم نظام التشغيل.

ترأس Matthew Felton الخبير في MS-DOS والذي عمل سابقاً على محيط توافق OS/2's MS-DOS، الفريق الذي أنشأ الأنظمة الفرعيّة MS-DOS و 16-bit Windows على النـظام Windows NT. يتعلّق المشروعان ببعضها ويشاركان مجموعة من الأهداف الكبير:

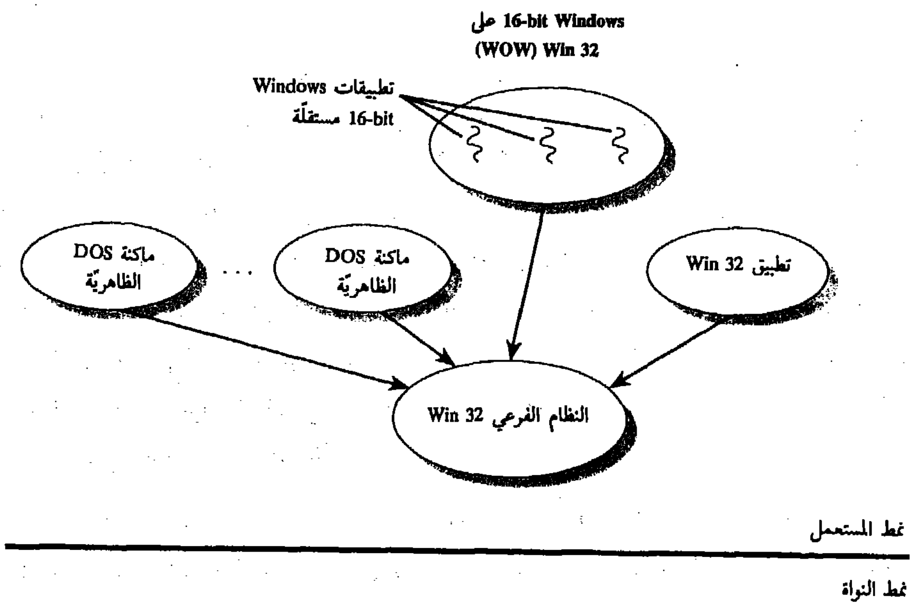
- الإتاحة للمستعملين الإنتقال بسهولة من MS-DOS أو 16-bit Windows إلى Windows NT.
- تشغيل كل تطبيقات MS-DOS و 16-bit Windows الرئيسيّة مع حماية بقيّة نظام التشغيل منها.

■ المحافظة على توافقيّة ثنائيّة للتطبيقات بين منصّات العتاد CISC و RISC.

■ الإتاحة لتطبيقات 16-bit Windows الإشتغال كنظير لتطبيقات 32-bit Windows.

على MS-DOS، فإن Windows هو تطبيق تخطيطي معقّد يمدّد قدرات نظام التشغيل الرئيسي وعلى Windows NT، فإن MS-DOS و 16-bit Windows هي تطبيقات: وهي أنظمة فرعيّة للمحيط والتي تستدعي روتينات Win 32 API وأحياناً خدمات NT المحليّة. يوضح الشكل (16-5) كيفيّة ملاءمة MS-DOS و 16-bit Windows في بنية النظام Windows NT.

يشتغل النظامان الفرعيان MS-DOS و 16-bit Windows في نمط المستعمل بنفس طريقة إشتغال الأنظمة الفرعيّة للمحيط الأخرى. لكن، وبعكس الأنظمة الفرعيّة Win 32 و OS/2 و POSIX، فهي ليست معالجات ملقّم، لأن تطبيقات MS-DOS تشتغل ضمن سياق معالجة تسمى ماكنة DOS الظاهريّة (VDM). إن VDM هي تطبيق Win 32 ينشيء حاسوباً ظاهرياً كاملاً يشغّل MS-DOS. فمثلاً، فإنه يتيح لتطبيقات MS-DOS إصدار تعليمات الماكنة، لاستدعاء BIOS للوصول مباشرة إلى أجهزة معيّنة وإستلام المقاطعات. يستطيع أي عدد من معالجات VDM الإشتغال في نفس الوقت، وكلٌّ ضمن إطار كونسول مستقلّ.



الشكل (16-5)

النظامان الفرعيان MS-DOS و 16-bit Windows

إن محيط 16-bit Windows هو تطبيق هجيني يشتغل ضمن فسحة عنوان معالجة VDM. وهو يستدعي روتينات Win 32 API للقيام بمعظم أعماله لكنّه يستدعي أحياناً خدمات NT. ويسمّيه مطوّروا محيط 16-bit Windows بالإسم WOW هو تعبير مختصر للتالي Windows On Win 32.

يوفّر إنشاء محيطات MS-DOS و16-bit Windows كأنظمة فرعيّة في نمط المستعمل لهذه المحيطات نفس حماية الشيفرة والبيانات التي تحتويها الأنظمة الفرعيّة الأخرى. وهي تحمي البرنامج التنفيذي NT من المشاكل التي يمكن أن تحصل في المحيطات لأنها تستطيع الوصول إلى البرنامج التنفيذي NT فقط باستدعاء خدمات النظام. تحمي هذه الإستراتيجيّة أيضاً تطبيقات MS-DOS من تطبيقات 16-bit Windows والعكس صحيح وهي تقسّم فسحات عناوينها عن تلك العائدة لتطبيقات 32-bit Windows.

1-4-5 ماكنات DOS الظاهريّة (VDMs):

VDM هي دورة MS-DOS تنشأ عندما يبدأ مستعمل تطبيق MS-DOS على Windows NT. يتيح النظام Windows NT تشغيل أي عدد من تطبيقات MS-DOS في نفس الوقت وهي تستطيع تمرير بيانات نصيّة إلى بعضها البعض وإلى تطبيقات Windows عبر الحافظة Clipboard.

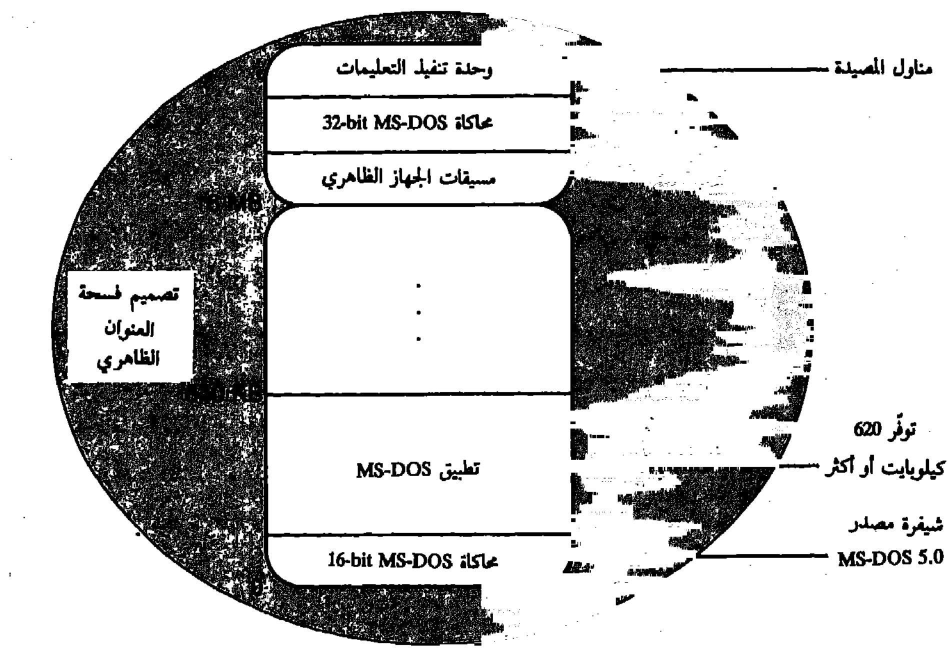
من الصعب تشغيل تطبيقات MS-DOS على Windows NT لأنها مكتوبة فعلياً بلغة assembly وهي تفترض أنها تتّصف بوصول حرّ إلى الذاكرة والأجهزة وما شابه. لكن في نظام تشغيل متعدّد المستعملين، لا يمكن لتطبيقات MS-DOS أن تكون حرّة، لكن يجب أن يُتاح لها الإشتغال على أنها كذلك. حقّق Sudeep Bharati و Dave Hastings، مطوّرا VDM الأوّلان، لقد قاما بهذه الذريعة بوضع كل تطبيق MS-DOS في معالجته الخاصة ـ VDM ـ مع فسحة عنوان ظاهري خاصة تحتوي كل شيفرة MS-DOS ومسيقات MS-DOS التي يحتاجها التطبيق للإشتغال. وضمن فسحة العنوان الظاهري، يستطيع التطبيق القيام بما يريد. يتحكّم برنامج إدارة الذاكرة الظاهريّة (VM) في البرنامج التنفيذي NT باستعمال الذاكرة الفعلي للتطبيق ويضمن عدم تجاوزها المعالجات الأخرى.

عندما ينقر المستعمل على رمز MS-DOS (أو على رمز تطبيق MS-DO) يبدأ النظام الفردي Win 32 غلاف أوامر Windows NT في إطار كونسول. يغادر غلاف الأوامر من نموذج OS/2 لتنفيذ أوامر 32-bit في غلاف واحد وأوامر MS-DOS في غلاف آخر. ورغم أنه يبدو كغلاف أوامر MS-DOS، فإنه يستطيع تنفيذ أوامر 32-bit Windows NT وأوامر 16-bit MS-DOS ضمن

نفس إطار الكونسول وحتى تمرير الخرج بين تطبيقات سطر الأمر. وعندما يدخل المستعمل أمراً، يستدعي غلاف الأمر الروتين () Win 32 CreateProcess لتنفيذ الرسم. وإذا كان الأمر رسم MS-DOS، يبدأ النظام الفرعي Win 32 معالجة VDM تحمّل تطبيق MS-DOS في فسحة عنوان الذاكرة VDM وتنفيذه. وعندما يولّد تطبيق MS-DOS الخرج تستدعي VDM روتينات كونسول Win 32 لعرض الخرج في إطار الكونسول.

عند إشتغاله، يجب أن يتمكّن تطبيق MS-DOS من الوصول إلى نظام التشغيل MS-Dos أو على الأقل إلى ما يشبه النظام MS-DOS ويعمل مثله. إن VDM هو نظام تشغيل MS-DOS ظاهري يشتغل على حاسوب ظاهري يعتمد على Intelx86. يظهر الشكل (17-5) تصميم فسحة العنوان الظاهري VDM على ماكنات تعتمد على Intelx86.

تشتقُّ محاكاة 16-bit MS-DOS من شيفرة مصدر MS-DOS 5.0 دون دعم نظام الملفّات. وهي تستقرُّ في القسم الأسفل من فسحة العنوان الظاهري في VDM مع وجود تطبيق MS-DOS مباشرة فوقها. ويستطيع التطبيق الوصول إلى 620 كيلوبايت من الذاكرة على الأقلّ.



الشكل (17-5)
ماكنة DOS الظاهريّة (VDM)

مع أن الشيفرة تحت حدود 16 ميغابايت تعتمد على عناوين مقطعيّة من 16 بت، تكتب الشيفرة فوق هذا الحدّ بالعناوين المسطّحة من 32 بت وهو نسق Windows NT. أما قسم 32 بت من فسحة عنوان VDM فهو معقّد. وهو يتضمّن مجموعة من مسيقات الجهاز الظاهري وشيفرة محاكاة 32-bit MS-DOS التي هي نفسها عبر بنيات معالج مختلفة. إن وحدة تنفيذ التعليمات هي مجموعة شيفرات تعتمد على المعالج. يعمل إصدار Intel x86، الذي كتب من قبل Dave Hastings من Microsoft، كمناول مصيدة (راجع الفصل السابع «النواة») حيث يلتقط التعليمات التي تؤدّي إلى إحتجاز العتاد ونقل التحكّم إلى الشيفرة التي تتناولها مثل مسيقات الجهاز الظاهري. وعلى معالجات MIPS، فإن هذه الشيفرة هي محاكي تعليمات، يحوّل تعليمات x86 إلى تعليمات MIPS.

تعمل مسيقات الجهاز الظاهري كطبقة بين تطبيقات MS-DOS والعتاد المثبّت بماكنة Windows NT. وفي إصداره الأول، يوفّر محيط VDM مسيقات جهاز ظاهري لأجهزة PC القياسيّة بما فيها الماوس ولوحة المفاتيح والطابعة ومنافذ COM وما شابه. وتتناول شيفرة 32-bit VDM عمليات دخل / خرج MS-DOS عن طريق إحتجازها واستدعاء إما وظائف Win 32 API أو البرنامج التنفيذي NT لتنفيذ عمليات الدخل / الخرج. فمثلاً، تعالج VDM طلبات منافذ COM عن طريق فتح مسيق جهاز COM وإرسال شيفرات التحكّم بالدخل / الخرج إليه (IOCTLs). ولتحديث الفيديو، تدقّق شعبة موجودة ضمن VDM دورياً ذاكرة الفيديو RAM حيث يكتب تطبيق MS-DOS ويستدعي روتينات API لكونسول Win 32 لتحديث عنصورة الشاشة التي تغيّرت.

مع أن إمكانيّة إشتغال عدّة دورات MS-DOS في نفس الوقت، يبقى إستعمالها للذاكرة قليل نسبياً. فأول 640 كيلوبايت من الذاكرة الظاهريّة في كل معالجة إضافة إلى أية ذاكرة تستعملها المعالجة حتى حدود 16 ميغابايت، هي فريدة وغير مشاركة ضمن VDM. وفوق هذا الحدّ، يشارك برنامج إدارة VM للبرنامج التنفيذي NT نسخة واحدة عن شيفرة 32-bit ضمن كل معالجات VDM. إضافة لذلك، ولأن VDM هي معالجات في نمط المستعمل، فهي قابلة لترتيب في صفحات. وهذا يعني أن برنامج إدارة VM في NT يحمّل في الذاكرة الفعليّة فقط تلك الأقسام من شيفرة MS-DOS 5.0 وشيفرة تطبيق MS-DOS التي يستعملها التطبيق، عند إستعمالها. وهو ينقل مؤقتاً محتويات ذاكرة التطبيقات إلى قرص إذا كان إستعمال الذاكرة في النظام مرتفعاً. (راجع الفصل 6 «برنامج إدارة الذاكرة الظاهريّة» للحصول على مزيد من المعلومات حول إدارة الذاكرة الظاهريّة).

إن تطبيقات MS-DOS ليست مهام متعدّدة لأن كل تطبيق يفترض أنه الوحيد الذي

يشتغل على ماكنة MS-DOS. لكن، مكوّن النواة NT يعامل شعبة MS-DOS كأية شعبة أخرى. وعندما تنتهي كميّة وقت الشعبة، يقاطعها النواة ويحوّل السياق إلى شعبة أخرى، حيث تتمّ إعادة جدولة شعبة MS-DOS لاحقاً. ولأن بعض تطبيقات MS-DOS تستقرُّ في حلقة محكمة تدقّق في دخل لوحة المفاتيح (وتستقلّ دورات CPU)، يكشف محيط VDM هذه الحالة المتوقّفة، وعندما تحصل، يوفّر للشعَب المنتظرة الأخرى أولويّة جدولة أعلى.

2-4-5 Windows on Win 32 (WOW):

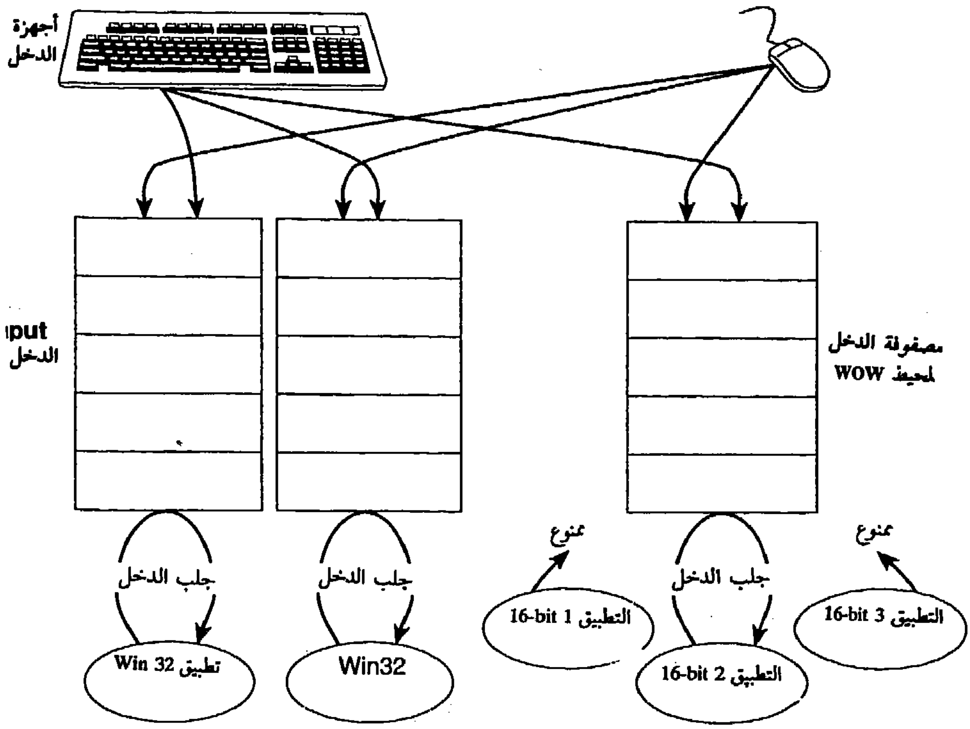
إحدى الأهداف الرئيسيّة لمحيط 16-bit Windows (WOW) كان إخفاء أي فرق واضح بين تطبيقات 16-bit Windows و 32-bit. فالمستعمل يبدأ تشغيل تطبيقات 16-bit Windows بنفس الطريقة التي يبدأ فيها تشغيل تطبيقات Win 32. ويشتغل كلا النوعين في نفس الوقت دون تمييز.

ورغم أن تطبيقات 16-bit و 32-bit تبدو هي نفسها للمستعمل، إلا أنها تشتغل فعلياً بظل تحكُّم الأجزاء المختلفة لنظام التشغيل. إن محيط WOW الذي صُمِّم واستُخدمَ من قبَل Jeff Parsons و Matthew Felton و Chandan Chauhan و Ramakrishna Nanduri هو ماكنة VDM متعدّدة الشعَب تنفذ كل من شعبها تطبيق 16-bit Windows. يحاكي تشغيل التطبيقات ضمن فسحة عنوان ظاهري واحد التصرُّف العادي للتطبيق 16-bit Windows، حيث كل التطبيقات أحادية الشعبة وتستقرُّ ضمن نفس فسحة العنوان. يستدعي محيط WOW روتينات Win 32 API لإنشاء الأُطر المعروضة على الشاشة وإدارتها لكل من تطبيقات 16-bit، وبالنسبة لدخل المستعمل، يعالج محيط WOW كتطبيق Win 32 أحادي، كما يبيّن في الشكل (5-18).

ومثل تطبيق MS-DOS، وفي أول مرّة يبدأ المستعمل تطبيق 16-bit Windows، يكتشف النظام الفرعي Win 32 أن الرسم المنفّذ يشتغل على MS-DOS ويبدأ معالجة VDM. وبعد بدئها، تحمّل VDM محيط WOW. تظهر فسحة العنوان الظاهري الماكنة WOW VDM في الشكل (5-19) على الصفحة 161.

تبيّن فسحة العنوان للنظام الفرعي WOW بشكل مشابه لتلك العائدة لمعالجة تطبيق MS-DOS. وتستقرُّ نفس شيفرة MS-DOS في القسم الأدنى من فسحة عنوان WOW مع وجود شيفرة النواة Windows 3.1 فوقها. تتناول شيفرة النواة مع إزالة دعم المهام المتعدّدة، وظائف إدارة الذاكرة Windows 3.1 وتحميل الرسوم المنفّذة ومكتبات الربط الديناميكي (DLL) لتطبيقات 16-bit Windows. ويقع برنامج إدارة النوافذ والروتينات الفرعيّة GDI فوق هذه الشيفرة وتستقرُّ تطبيقات 16-bit Windows فوق الروتينات الفرعيّة. ويمكن تشغيل أي عدد من التطبيقات هناك

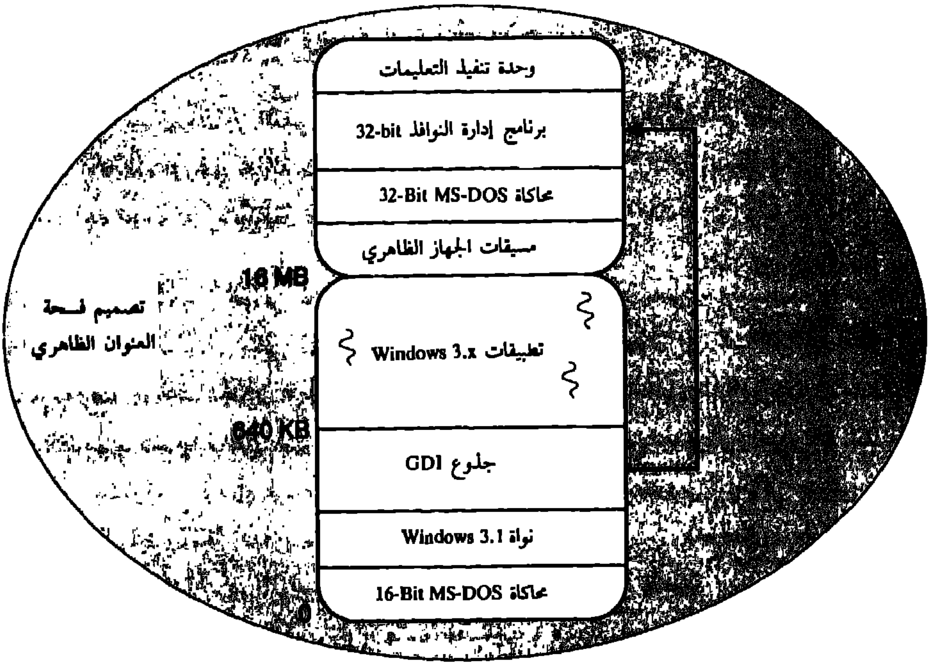
وشيفرتها والبيانات المرتّبة في صفحات في الذاكرة بواسطة برنامج إدارة VM في NT عند وصول التطبيقات إليها.



الشكل (5-18)
نموذج الدخل لمحيط WOW

في النظام الفرعي WOW، تستبدل شيفرة المهام المتعدّدة 16-bit Windows بشيفرة WOW بواسطة الإستدعاءات إلى روتينات Win 32 API وبواسطة شيفرة المهام المتعدّدة للبرنامج التنفيذي NT. ويعد إشتغال محيط WOW، يرسل النظام الفرعي Win 32 رسالة إليه في كل مرة يبدأ المستعمل تطبيق 16-bit. يستجيب WOW بتحميل التطبيق في الذاكرة وإستدعاء روتين Win 32 CreateThread () API لإنشاء شعبة لتشغيل التطبيق. ورغم جدول كل بقيّة الشعَب شفعياً في Windows NT، يجدول النظام الفرعي Win 32 شعَب WOW غير شفعياً ليجعل محيط WOW متوافقاً مع 16-bit Windows. وهذا لا يعني أنه يُتاح لشعَب WOW التشغيل لكميّة الوقت التي تريد. فالنواة NT ما تزال تقاطع تنفيذ شعبة WOW لتتيح تشغيل الشعَب غير العائدة للنظام الفرعي WOW. ولكن، عندما تحوّل مجدّداً إلى WOW، تنتقي النواة فقط شعبة WOW

المقاطعة لمتابعتها. وتبقى كل شعَب WOW الأخرى ممنوعة إلى أن تتخلى إحداها عن المعالج. يتوازى هذا التصرُّف مع المهام المتعدّدة غير الشفعيّة التي تتوقّعها تطبيقات Windows 3.X دون التأثير على Win 32 أو أية تطبيقات أخرى تشغيل على Windows NT.



الشكل (19-5) 16-Bit Windows
على Win 32 (WOW)

يوجد فوق حدود 16 ميغابايت في فسحة عنوان النظام الفرعي WOW نفس الشيفرة الموجودة في معالجات التطبيق MS-DOS مسيقات جهاز MS-DOS ومحاكاة 32-bit MS-DOS ووحدة تنفيذ التعليمات المعتمدة على العتاد. إضافة لذلك، يوجد كتلة من برنامج إدارة الأُطر 32-bit وشيفرة GDI التي تشبه شيفرة 16-bit في القسم الأدنى من فسحة العنوان. تكون شيفرة 32-bit هذه مسؤولة عن ترجمة العناوين المقطعيّة 16-bit إلى عناوين مسطّحة 32-bit. فمثلاً، عندما يستدعي تطبيق 16-bit Windows برنامج إدارة الأُطر أو وظائف GDI، تستدعي جذور 16-bit المبيّنة في الشكل 19-5 وظائف 32-bit API المعادلة في القسم الأعلى من فسحة عنوان النظام الفرعي WOW. ويأخذ برنامج إدارة الأُطر 32-bit وشيفرة GDI عناوين 16-bit المزوّدة من قبل التطبيق وتعدّلها لتوافق نموذج العنونة المسطحة 32-bit. ثم يستدعي روتين Win 32 API

لتنفيذ العمليّة. وعندما يرجع النظام الفرعي Win 32 نتائجه، تعدل شيفرة 32-bit WOW عناوين 32-bit مجدداً إلى عناوين مقطعيّة 16-bit وترجع النتائج إلى التطبيق. ولأن روتين 16-bit API غير مستخدم فعلياً في WOW، فإن أي تطبيق 16-bit Windows يعتمد على البنية الداخلية لبرنامج إدارة الأطر 16-bit أو GDI لا يضمن أن تعمل على Windows NT.

5-5 تمرير الرسائل بواسطة وسيلة إستدعاء الإجراء المحلّي (LPC):

في نموذج المستضاف / الملقّم في Windows NT، يتعلّق الكثير على نجاح وسيلة إستدعاء إجراء محلّي (LPC). ورغم محاولة كل نظام فرعي بذل جهده لعدم إرسال رسائل LPC (راجع القسم 5-1-2)، يجب على الأنظمة الفرعيّة القيام بذلك في بعض الأوقات.

عند تواجد شعبتين ضمن نفس المعالجة، فإنها تشارك فسحة عنوان وتستطيع الإتصال بالبيانات وتمريرها بسهولة. وهي تستعمل آليات مزامنة بسيطة للوصول إلى البيانات في التتابع الصحيح. وعند تواجد شعبتين في معالجات مختلفة، فإنها يجب أن توصل الفجوة بين فسحات العنوان الظاهري المستقلّة عن طريق نسخ البيانات من فسحة عنوان واحدة إلى أخرى أو بإنشاء منطقة ذاكرة مشاركة مرئيّة في فسحتي العنوان. إن LPC هي وسيلة لتمرير الرسائل متوفّرة من قبل البرنامج التنفيذي NT. وهي تستعمل في الحالات الأخيرة ـ أي بين معالجتين، ومستضاف ونظام فرعي محمي (ملقّم)، الموجودة على نفس الحاسوب. يحاكي تصميم وسيلة LPC نموذج إستدعاء الإجراء المستعمل من قبل المواصفات الصناعيّة، وسيلة إستدعاء إجراء بعيد (RPC) المستعمل لتمرير الرسائل بين معالجات المستضاف والملقّم على حواسيب مختلفة. وفي وسيلة RPC، لا يعرف التطبيق الذي يرسل الرسالة أنه يمرّر رسالة. فهو يستدعي روتين API يشبه أي روتين API آخر ويعاود إجراء جذور توضيب البارامترات إلى الروتين ويستدعي وسيلة RPC لإرسالها إلى ملقّم بعيد. وترجع النتائج عبر نفس القناة. (راجع الفصل التاسع «إنشاء الشبكات» لمزيد من المعلومات).

تعمل وسيلة LPC في Windows NT كوسيلة RPC لكنها مستمثلة لمعالجتين تشتغلان على نفس النظام Windows NT. يستدعي تطبيق روتين API في مكتبة DLL حيث يربط وتقوم المكتبة DLL بكل ما هو مطلوب لإرسال الرسالة إلى نظام فرعي محمي Windows NT. ورغم أن وسيلة RPC هي آلية عامة مستعملة على أنواع مختلفة من أنظمة التشغيل، تحدّد وسيلة LPC للنظام Windows NT وبالتالي تستغلّ مزايا Windows NT لجعلها أسرع وأكثر كفاية من وسيلة RPC العامة.

تزوّد وسيلة LPC للبرنامج التنفيذي ثلاث طرُق مختلفة لتمرير الرسائل، كل منها مصمّم لحالة مختلفة:

- إرسال رسالة إلى كائن منفذ متعلّق بمعالجة ملقّم.
- إرسال مؤشّر رسالة إلى منفذ ملقّم وتمرير الرسالة في ذاكرة مشاركة.
- تمرير رسالة إلى شعبة ملقّم معيّن عبر منطقة ذاكرة مشاركة محدّدة.

إضافة لذلك، تزوّد وسيلة LPC آلية معاودة إستدعاء معقّدة تتيح للملقّم الإجابة على رسالة بطلب مزيد من المعلومات حول المستضاف.

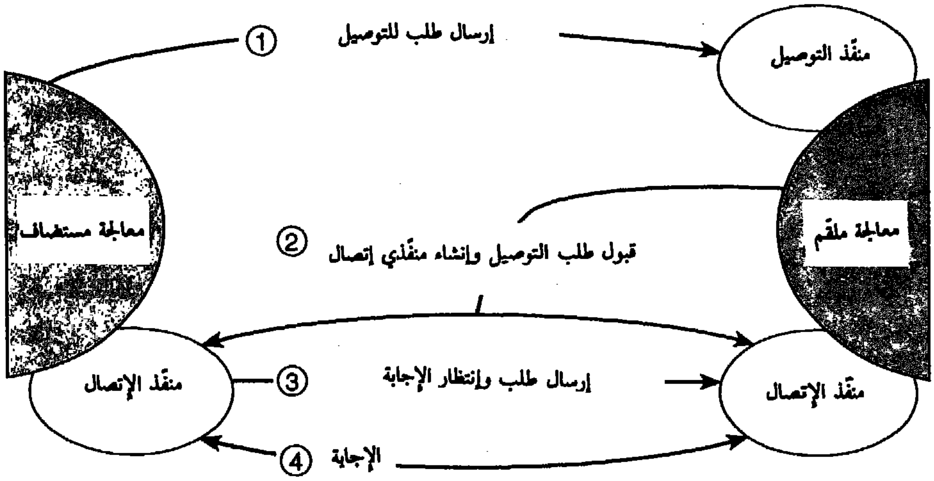
يعالج القسم الفرعي التالي كيفيّة قيام معالجة مستضاف بإنشاء توصيل مع معالجة ملقّم باستعمال كائن منفّذ. يعالج القسم الفرعي اللاحق الأنواع المختلفة لتمرير رسالة LPC ومعاودة إستدعاء LPC بتفصيل أكبر.

1-5-5 كائن المنفّذ:

في كل أشكال تمرير الرسالة LPC، يجب أن تحقّق معالجة مستضاف قناة إتصال مع نظام فرعي محمي قبل إرسال رسالة إلى النظام الفرعي. يستعمل البرنامج التنفيذي NT مثل نظام التشغيل Mach، كائناً منفّذاً كوسيلته لإنشاء توصيل والمحافظة عليه بين معالجتين.

يستطيع أي عدد من المستضافات إستدعاء نظام فرعي محمي. وبالتالي يحتاج كل منها لقناة إتصال آمنة وخاصة. ولإستيعاب هذه الحاجة، يستخدم البرنامج التنفيذي NT نوعين من المنافذ. وتتشابه بنياتها لكنها تختلف في أسمائها العامة وفي كيفيّة إستعمالها من قبل NT. أحد أنواع المنافذ يسمى منفّذ التوصيل وهو يوفّر لتطبيقات المستضاف مكان إستدعاء لاعداد قناة إتصال مع الملقّم. تحتوي كائنات منفّذ التوصيل على أسماء تجعلها مرئيّة لكل معالجات NT، يوضح الشكل (5-20)، خطوة بخطوة، كيفيّة تحفيز معالجة مستضاف للإتصال مع نظام فرعي محمي وكيفيّة إرسال الرسائل اللاحقة.

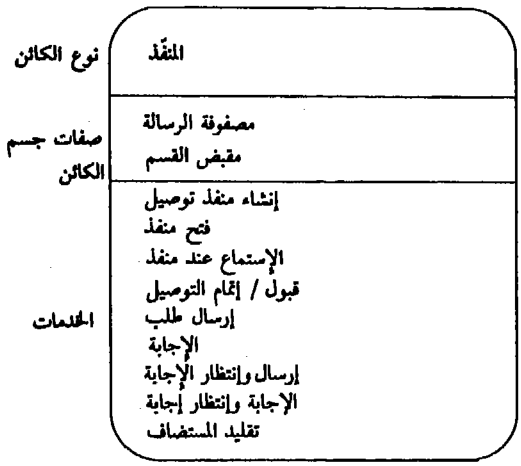
لتحفيز إتصال مع نظام فرعي محمي، تفتح معالجة مستضاف مقبض إلى كائن منفّذ طويل للنظام الفرعي المحمي ثم يرسل له طلب توصيل. يستجيب الملقّم، الذي يحتوي على شعبة واحدة أو أكثر تنتظر إستلام مثل هذه الطلبات، وذلك بإنشاء كائني منفّذ إتصال دون أسماء (وبالتالي خاصة)، مع إبقاء مقبض واحد وإرجاع المقبض الآخر إلى المستضاف. يستعمل المستضاف مقبض منفّذ الإتصال لإرسال الرسائل اللاحقة أومعاودة الإستدعاء إلى النظام الفرعي المحمي والإستماع إلى الإجابات من النظام الفرعي. يستعمل النظام الفرعي مقبضه بنفس الطريقة للإتصال مع المستضاف.



الشكل (5-20)

التوصيل والإتصال مع نظام فرعي محمي

يلخّص الشكل (5-21) صفات وخدمات NT المحليّة لمناولة كائنات المنفّذ.



الشكل (5-21)
كائن المنفذ

لا يمكن تأصُّل المقابض إلى كائنات المنفذ، عكس معظم مقابض كائنات NT الأخرى، من قبل معالجة حديثة الإنشاء. وإذا سمح التأصيل، يجب على الملقّم معرفة المعالجة التي تستدعيه في كل مرة يستلم رسالة. وبواسطة منع التأصيل، يعرف الملقّم دائماً المعالجة التي تستدعيه على قناة معيّنة، وبالتالي يخفض كلفات معالجة الملقّم وتجعل المستضافات على خدمة أسرع.

2-5-5 أنواع تمرير رسائل LPC:

عندما تنشىء معالجة مستضاف قناة إتصال مع نظام فرعي محمي، فإنه يحدّد نوعاً من الأنواع الثلاثة لطرق تمرير رسالة LPC التي تريد إستعماله:

- تمرير الرسائل إلى مصفوفة رسائل كائن المنفذ هي طريقة مستعملة للرسائل الصغيرة.
- تمرير الرسائل عبر كائن ذاكرة مشاركة هي طريقة مستعملة للرسائل الكبيرة.
- تستعمل LPC السريعة بشكل خاص من قبل أقسام من النظام الفرعي Win 32 لتحقيق كلفة دنيا وسرعة قصوى.

إن هذه الأنواع الثلاثة لتمرير الرسائل وآلية معاودة الإستدعاء هي مواضيع الأقسام الفرعيّة التالية.

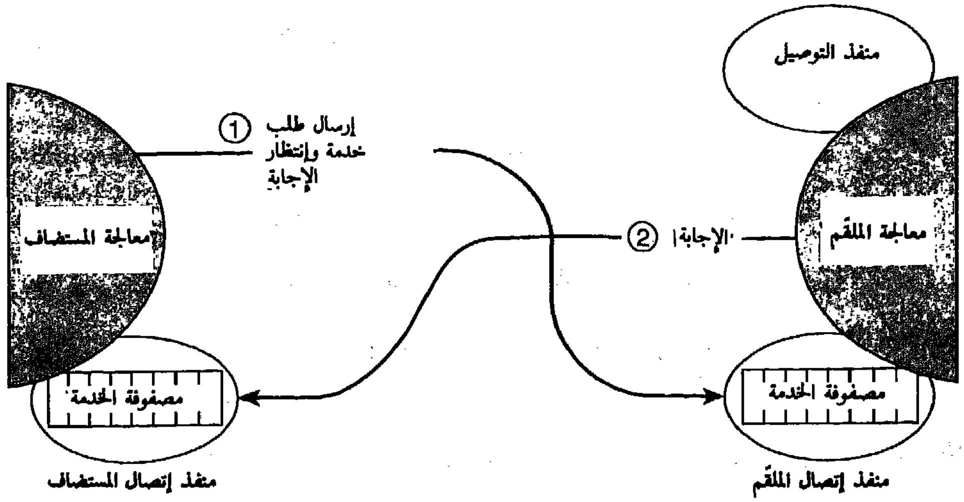
1-2-5-5 نسخ الرسالة إلى مَنْفذْ:

يربط أول وأكثر الطرق العامة لتمرير الرسائل فسحات عنوان المستضاف والنظام الفرعي المحمي عن طريق نسخ رسالة مستضاف إلى موقع متوسط ثم نسخه إلى فسحة عنوان النظام الفرعي. والموقع المتوسط الذي يستعمله هو مصفوفة رسائل في كائن منفذ الإتصال.

كما يظهر الشكل (5-21)، يحتوي كل كائن منفذ على مصفوفة من كتل رسائل ثابتة القياس. (تحتوي كائنات منفذ الإتصال على مصفوفة لطلبات التوصيل وتحتوي كائنات منفذ التوصيل على مصفوفة لطلبات الملقّم). وعندما يرسل مستضاف رسالة، تنسخ وسيلة LPC هذه الرسالة في إحدى كتل الرسائل في كائن منفذ النظام الفرعي. وبعد أن يحوّل سياق النواة NT من المستضاف إلى معالجة النظام الفرعي، تنسخ شعبة نظام فرعي الرسالة إلى فسحة عنوان النظام الفرعي وتعالجها. وعندما يريد النظام الفرعي الإجابة، فإنه يرسل رسالة إلى منفذ إتصال المستضاف، كما يظهر في الشكل (5-22).

وفي أي وقت، يستطيع البرنامج التنفيذي NT الوصول إما إلى فسحة عنوان المستضاف أو إلى فسحة عنوان النظام الفرعي، لكن ليس إلى كلاهما. وكسائر الكائنات الأخرى، تخزّن كائنات المنفذ في ذاكرة النظام، بحيث لا يفقد الوصول إلى الرسالة عندما يحوّل سياق النواة NT من معالجة المستضاف إلى معالجة النظام الفرعي.

عند إنشاء كائن منفذ، تحدّد وسيلة LPC موقع ذاكرة له من المجموعة غير المرتّبة بصفحات. أي، ذاكرة النظام المتواجدة دائماً. ولأن المجموعة غير المرتبة بصفحات هي مورد



الشكل (5-22)
وسيلة LPC لنسخ الرسائل

نظام محدود، تحد كتل الرسائل في كائن منفذ لجهة القياس وفي العدد. يبلغ قياس كتلة الرسائل 256 بايت، وهو ما يكفي لإرسال معظم الرسائل العادية.

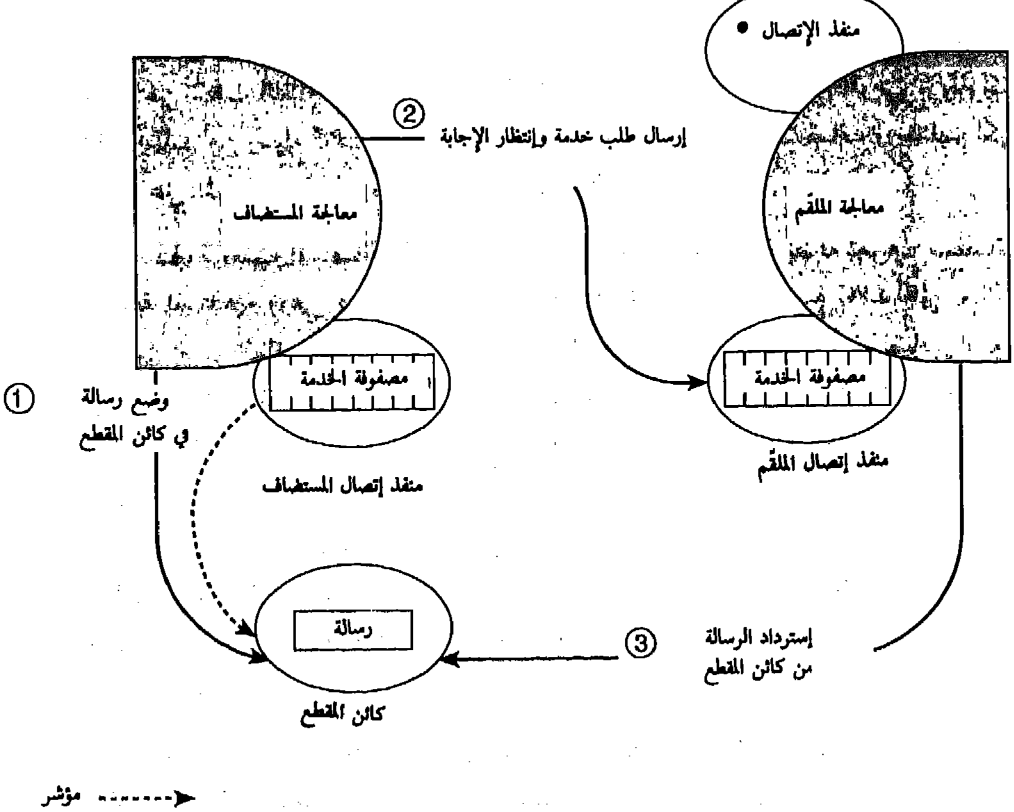
5-5-2-2 تمرير رسالة في الذاكرة المشاركة:

عندما يمرّر مستضاف رسائل أكبر من 256 بايت، فإنه لا يستطيع نسخها إلى مصفوفة رسائل منفذ الملقّم. وعِوَضاً عن ذلك، يجب أن يمرّرها عبر كائنات الذاكرة المشاركة والتي يحدّ حجمها وفقاً لحدود حصص موارد المستضاف.

لتمرير الرسائل باستعمال الذاكرة المشاركة، ينشىء المستضاف كائن ذاكرة مشاركة، تسمى كائن مقطع. وكائن المقطع (الذي يوصف بتفصيل أكبر في الفصل السادس «برنامج إدارة الذاكرة الظاهريّة») هو كتلة ذاكرة مشاركة تجعلها وسيلةLPC مرئيّة في نسختي عنوان المستضاف والنظام الفرعي المحمي. ولإرسال رسالة كبيرة، يضعها المستضاف في كائن المقطع ثم يرسل إلى منفذ الملقّم رسالة صغيرة تحتوي مؤشّراً ومعلومات حجم الرسالة الكبيرة. وبعد أن يحوّل سياق النواة NT إلى معالجة النظام الفرعي، يسترد النظام الفرعي المعلومات من كتلة الرسالة ثم يستعملها لإيجاد الرسالة في كائن المقطع، كما يوضّح ذلك الشكل (5-23).

لاحظ أنه يجب على المستضاف تقرير معنى إنشاء لأول مرة قناة إتصال وإذا كانت رسائله كبيرة أو صغيرة. فإذا كان يتوقّعها صغيرة، فإنه لا يطلب كائن مقطع، لكن إذا توقع على الأقل

أن تكون إحدى رسائله كبيرة، فإنه يطلب كائن المقطع. وكما يظهر الشكل، يتعلق كائن المقطع مع منفذ إتصال المستضاف. وإذا توقّع النظام الفرعي أيضاً أن تكون رسائل الإجابة كبيرة أيضاً، فإنه ينشىء كائن مقطع متعلّق بمنفذ الإتصال الخاص به الذي يخزّن الإجابات الكبيرة عند إرسالها إلى المستضاف.



الشكل (5-23)
وسيلة LPC للذاكرة المشاركة

إن إستعمال كائن مقطع يعني أنه يجب على المستضاف القيام بعمل إضافي. فوسيلة LPC لا تفترض أي نسق معينّ لكائن المقطع، مثلاً، لكي يستطيع المستضاف إدارة ذاكرة كائن المقطع نفسه، حيث يبلغ النظام الفرعي عن حجم الرسالة الأخيرة وموقع تخزينها ضمن كائن المقطع. إن عمليّة إدارة الذاكرة هذه تجعل من تمرير الرسائل عبر الذاكرة المشاركة أكثر تعقيداً للمستضاف مقارنة مع إستعمال كائن منفذ مباشرة. ولكنها تتيح إستيعاب الرسائل الكبيرة دون

نسخها عدّة مرات. وقد تكون عملية نسخ كميات كبيرة من البيانات من فسحة عنوان واحد إلى آخر عملية بطيئة وبالتالي يؤدّي إستعمال الذاكرة المشاركة إلى تجنُّب كلفة المعالجة الإضافيّة هذه.

3-2-5-5 معاودة الإستدعاء:

بظل الحالات العاديّة، يحتوي النظام الفرعي عادة على عدّة منافذ إتصال. ويستعمل كل منها كقناة إتصال لمعالجة مستضاف واحد. ولخدمة الطلبات التي تستلمها على منافذ الإتصال المتعدّدة، ينشىء النظام الفرعي مجموعة شعَب تنتظر إستلام الطلبات ومعالجتها. وتستطيع أية شعبة من شعَب النظام الفرعي الإجابة على أي طلب. وهذا يوفّر للنظام الفرعي مرونة أكبر لكنّه يطلب من وسيلة LPC الإحتفاظ بمخطط جيد لتعريف مستدعي المستضاف ورسائلهم لكي تجيب على المستضاف الصحيح عند منفذ الإتصال الصحيح.

لمتابعة تعقّب المستضاف الذي يرسل رسائل، تحتوي كل رسالة بطاقة تعريف مستضاف الشعبة المستدعية (وهي صفة لكل كائن شعبة) ورقم تسلسلي تُعَيِّنُهُ وسيلة LPC لكل رسالة. وعندما يجيب النظام الفرعي على رسالة، يسجل في إجابته بطاقة تعريف مستضاف الشعبة والرقم التسلسلي للرسالة التي يجيب عليها. بعد ذلك تتأكد وسيلة LPC من أن المستضاف ذات بطاقة تعريف المستضاف ينتظر إجابة لرقم الرسالة هذا. وإذا لم يكن كذلك، ترجع وسيلة LPC رسالة خطأ.

في بعض الأحيان، قد لا يتمكّن النظام الفرعي من إرسال إجابة فوراً. وقد يطلب معلومات إضافيّة من المستضاف. توفّر وسيلة LPC آلية معاودة الإستدعاء لاستيعاب هذه الحالة. يوضح الشكل (5-24) على الصفحة التالية تبادل نموذجي. (الخطوة 2 هي رسالة معاودة الإستدعاء).

يرسل مستضاف عادة طلباً ثم ينتظر إجابة. لكن إذا كان المستضاف يدعم مزيّة معاودة الإستدعاء، فإنه يستطيع مناولة طلب من النظام الفرعي عندما يكون منتظراً إجابة. وبإستعمال روتينات LPC المحلية، يستطيع المستضاف الإجابة على الطلب ثم مواصلة إنتظار الإجابة الأصلية.

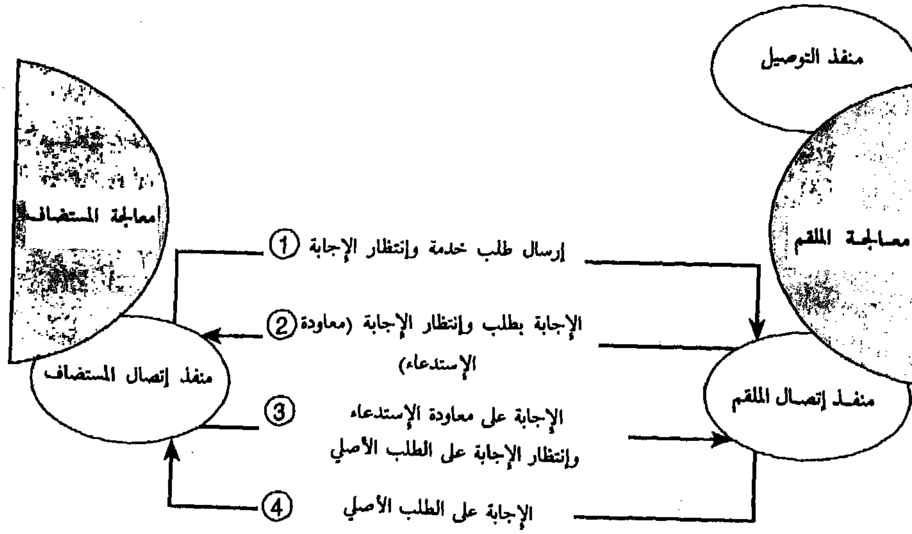
لا تحتوي وسائل تمرير الرسائل على العديد من أنظمة التشغيل مرونة البرنامج التنفيذي NT في إستخدام معاودة الإستدعاء. فمثلاً، فإن آلية معاودة الإتصال LPC في NT متناظرة تماماً. ويستطيع المستضاف والملقّم إصدار معاودة إتصال إلى بعضها البعض. إضافة لذلك، تتيح وسيلة LPC لعدد غير محدّد من معاودات الإستدعاء بالتواجد في نفس الوقت. في الخطوة 3 في الشكل (5-24)، مثلاً، وعِوَضاً عن الإجابة على معاودة إستدعاء الملقّم، يطلب

المستضاف معلومات إضافيّة من الملقّم تتعلّق بمعاودة إستدعاء الملقّم. فإذا حصل ذلك، يرسل المستضاف معاودة إستدعاء إلى الملقّم ثم ينتظر إجابة لمعاودة الإستدعاء وإجابة للطلب الأصلي. بمعنى آخر، يمكن تكرار الخطوة 2 عدّة مرات إما بواسطة المستضاف أو بواسطة الملقّم، ويمكن أن تتكدّس كل معاودات الإستدعاء التي لم تتمّ الإجابة عليها على الجانبين. وتتمّ الإجابة على كل معاودة إستدعاء واحد بعد الآخر إلى أن تتمّ الإجابة على كل معاودات الإتصال على الجانبين. وعند حصول ذلك، يرسل الملقّم أخيراً إجابة على الطلب الأصلي للمستضاف وتتبادل الرسالة الطرفين.

4-2-5-5 وسيلة LPC السريعة:

كما يلاحظ فإن Win 32 هو نظام فرعي عالي الإستعمال. ولأنه يتفاعل مع كل التطبيقات الشغّالة على Windows NT، يمكن الحصول على عدّة إستدعاءات في أي وقت. ولأن محاولة النظام الفرعي أن يكون في مقدّمة الأداء، إستخدم مطوِّروه Mark Lucovsky و Steve Wood، وسيلة LPC السريعة وهي شكل مستمثل لتمرير رسائل LPC للإستعمال من قبل النظام الفرعي Win 32. يستعمل برنامج إدارة الأطر في النظام الفرعي Win 32 ومكوّن GDI وسيلة LPC السريعة لتخفيض الوقت المطلوب لإستدعاء LPC بإتجاهين من مستضاف إلى ملقّم Win 32.

إن معظم الكلفة الإضافيّة المشمولة في إستعمال وسيلة تمرير رسائل LPC هي في فتح



الشكل (5-24)
معاودة الإستدعاء

مقبض إلى كائن منفذ ونسخ الرسائل إلى ومن مصفوفة الرسائل. وحتى عندمـا يستعمل مستضاف وسيلة LPC للذاكرة المشاركة، فإنه يرسل رسالة إلى مصفوفة الرسائل الواجب نسخها إلى فسحة عنوان النظام الفرعي.

في وسيلة LPC السريعة، ترسل شعبة مستضاف رسالة إلى منفذ توصيل الملقّم لإنشاء إتصال، حيث تشير إلى رغبتها في إستعمال وسيلة تمرير رسائل LPC السريعة للإتصال. إستجابة لذلك، ينشيء الملقّم ثلاثة موارد للمستضاف:

- شعبة ملقّم محدّدة لمناولة هذه الطلُب والطلبات اللاحقة.
- 64 كيلوبايت من الذاكرة لمشاركة (كائن مقطع) لتمرير الرسائل.
- كائن زوج أحداث.

بعد ذلك، تقوم شعب المستضاف والملقّم بإستعمال وسيلة LPC السريعة بتجاوز الأُطر وتمرير رسائلها عبر الذاكرة المشاركة. ويوفّر كائن زوج الأحداث، الذي يحتوي حدثاً واحداً لشعبة المستضاف وحدثاً واحداً لشعبة الملقّم، آلية مزامنة ضمنيّة. فعلى سبيل المثال، يضع المستضاف رسالة في كائن المقطع ثم يضبط حدث الملقّم مع إنتظار الحدث الخاص به. وتحفز النواة NT شعبة الملقّم المحددة، ولأن الوظيفة الوحيدة للشعبة هي خدمة شعبة مستضاف واحدة، فإنها تعرف فوراً موقع البحث عن الرسائل. عندما تنتهي شعبة ملقّم من معالجة الرسالة، فإنها تضبط حدث المستضاف وتنتظر في نفس الوقت الحدث الخاص بها. يتواصل الإتصال بهذه الطريقة إلى أن يتم إغلاق توصيلة LPC السريعة.

وبإزالة الكلفة الإضافيّة لإستعمال كائن المنفذ والكلفة الإضافيّة لنسخ الرسائل بين فسحتي عنوان المعالجتين، توفّر وسيلة LPC السريعة للنظام الفرعي Win 32 قوّة في الأداء. إضافة لذلك، ولأن شعبة ملقّم محدّدة أُنشئت لخدمة أية شعبة مستضاف، يتجنّب النظام الفرعي محاولة معرفة شعبة المستضاف التي تحاول إستدعاءه في كل مرة يستلم رسالة. وبإستعمال وسيلة LPC السريعة، يصبح تحويل السياق من المستضاف إلى النظام الفرعي (والعكس بالعكس) عامل إزالة الحدود في أداء تمرير الرسائل وتخفّض النواة NT حتى ذلك بتخصيص أفضلية لهذه الشعَب في الجدولة.

لذلك لماذا حد إستعمال وسيلة تمرير رسائل LPC السريعة إلى النظام الفرعي Win 32؟ فإذا كانت سريعة جداً، لماذا لا تستعمل هذه الطريقة لكل أشكال تمـرير الرسائـل على Windows NT؟ الجواب هو أن وسيلة LPC السريعة تبادل نوعاً واحداً من الكلفة الإضافيّة للنظام بآخر. وكل ما تكسبه بالسرعة تخسره في إستعمال الموارد. وعِوَضاً عن المحافظة على

مجموعة شعَب، حيث كل منها يستجيب لعدّة مستضافات، تتطلّب وسيلة LPC السريعة إنشاء شعبة ملقّم لكل شعبة مستضاف تستدعيها. وإضافة إلى إستهلاك ذلك لنظام الذاكرة (البعض منها ذاكرة موجودة غير مرتّبة بصفحات)، تمضي هذه الشعَب المحدّدة نصف وقتها بإنتظار تحفيزها من قبَل المستضاف، وتكمن كلفة الموارد الأخرى في إستعمال كائن المقطع. ففي وسيلة LPC العادية، تستطيع كل معالجة إنشاء كائن مقطع لتمرير الرسائل. وإذا مررت أكثر من شعبة واحدة الرسائل، تشارك الشعَب كائن مقطع. في وسيلة LPC السريعة، تحتوي كل شعبة على كائن مقطع خاص بها.

تخفّف بعض كلفات الموارد هذه بالواقع أن الشعَب تكدّس وذاكرة كائن المقطع مرتّبة بصفحات ويمكن نقلها إلى قرص عندما يكون إستعمال الذاكرة عالياً. لكن لن يكون عملياً إستعمال وسيلة LPC السريعة لكافة تمريرات الرسائل. فوسيلة LPC السريعة تستعمل فقط من قبَل برنامج إدارة النوافذ Window Manager ومكوّنات GDI للنظام الفرعي Win 32، بينما يستعمل الكونسول ومكوّنات نظام التشغيل وأيضاً كل الأنظمة الفرعيّة في Windows NT، وسيلة LPC العاديّة.

6-5 بإختصار:

نموذج المستضاف / الملقّم هو الجزء الأساسي من تصميم Windows NT، وهو يؤثّر، كيفيّة تشغيل التطبيقات إضافة إلى تأثيره على كيفيّة عمل النظام. ولقد تمّ إختياره في بادىء الأمر لمرونته في توفير روتينات API للأنظمة OS/2 و POSIX ضمن نفس نظام التشغيل لكنه أصبح أساس النظام الفرعي المحمي Win 32 أيضاً. بإستعمال نموذج المستضاف / الملقّم، تتواجد تطبيقات MS-DOS و 16-bit Windows سويّة مع تطبيقات Win 32 و OS/2 و POSIX، وتستطيع كل التطبيقات تمرير البيانات إلى بعضها البعض عبر الحافظة Clipboard. كذلك يحمي نموذج المستضاف / الملقّم التطبيقات المختلفة من بعضها البعض والأنظمة الفرعيّة المحميّة من التطبيقات.

عالج هذا الفصل كيف يبدو Windows NT للمستعملين وللبرامج التطبيقيّة. ويعالج الفصل التالي الأعمال الأساسيّة للبرنامج التنفيذي NT وبالتحديد برنامج إدارة VM. رغم أن الأنظمة الفرعيّة للمحيط المختلفة تقدّم مشهد الذاكرة الذي تتوقّعه التطبيقات، يقع تحت الأنظمة الفرعيّة نظام الذاكرة الظاهريّة للبرنامج التنفيذي NT. فموضوع الفصل التالي هو برنامج إدارة الذاكرة الظاهريّة ومكوّن NT الذي يمنع كل التطبيقات والأنظمة الفرعيّة المحميّة من الإصطدام ببعضها البعض.

6

برنامج إدارة الذاكرة الظاهريّة

فيما مضى كانت الحواسيب عبارة عن أنظمة آحاديّة المعالجة وآحاديّة الشعبة. وكان يتوجّب على المبرمجين الهواة والأخصّائيّين تسجيل وقت للعمل على كونسول واحد للحاسوب. فالمبرمج كان نظام التشغيل المسؤول عن تحميل برنامج يدوي في الذاكرة بإستعمال المفاتيح والشريط الورقي أو البطاقات المخرّمة. وبعد تحميل البرنامج، كان المبرمج يدخل عنوان البدء ويوجّه المعالج ليقفز إليه والبدء بالتنفيذ. ولقد كان من المستحيل تحميل وتنفيذ أكثر من برنامج واحد في كل مرّة. ولقد بقي المعالج دون عمل لفترات طويلة.

وفي تقنيّة نظام التشغيل، فقد أوجد التقدّم وسائل إبقاء المعالج مشغولاً لفترات أطول وبالتالي تنفيذ أعمال أكثر. تحمل أنظمة المهام المتقدّمة عدّة برامج في الذاكرة وإبقاء المعالج مشغولاً بالتبديل فيما بينها. أما كيفيّة توزيع نظام التشغيل الذاكرة المتوفّرة على المعالجات خلال حماية شيفرة وبيانات معالجة واحدة من المعالجات الأخرى، فهي موضوع إدارة الذاكرة، وفي حالة Windows NT، إدارة الذاكرة الظاهريّة.

في أول أيام الحواسيب، كان من المتعذّر تنفيذ برنامج أكبر من الذاكرة الفعليّة للحاسوب. ولاحقاً، بدأ المبرمجون بكتابة برامج متراكبة تبادل أجزاءاً من شيفرتها إلى قرص وتحمّل أجزاء البرنامج الأخرى إلى الذاكرة. وعند الحاجة للشيفرة الموجودة على القرص، يعاود البرنامج تلقيمها إلى الذاكرة، حيث يضع الشيفرات في طبقات على تلك غير المستعملة. وإلى جانب صعوبة البرمجة والتحديث، كانت تتطلّب عمليّة التراكب الطبقي من كل تطبيق إعادة إنشاء الشيفرة التي تبادل محتويات الذاكرة إلى قرص.

الذاكرة الظاهريّة (VM) التي استخدمت لأول مرة في العام 1959، أخذت على عاتق إدارة الذاكرة من المبرمج ووضعته على نظام التشغيل. إن VM هو نظام مركزي لمبادلة محتويات الذاكرة إلى قرص عندما تمتلىء الذاكرة. وهي تتيح للمبرمجين إنشاء وتشغيل البرامج التي تتطلّب ذاكرة أكبر مما هو متوفّر على الحاسوب وأصبحت طريقة إدارة الذاكرة القياسيّة لكل أنظمة التشغيل ما عدا البسيط منها.

إن مكوّن الذاكرة الظاهريّة في البرنامج التنفيذي NT، برنامج إدارة VM، هو نظام إدارة الذاكرة المحليّة للنظام Windows NT. تعتمد قدرات إدارة الذاكرة التي يوفّرها النظام الفرعي للمحيط على برنامج إدارة NT's VM. وقد صمّم برنامج إدارة VM واستخدم من قبَل Lou Perazzoli الذي كان أيضاً مدير قسم الهندسة ومدير مشروع NT. بالإضافة إلى أفراد الفريق الآخر، حقّق Lou أهداف برنامج إدارة VM التالية:

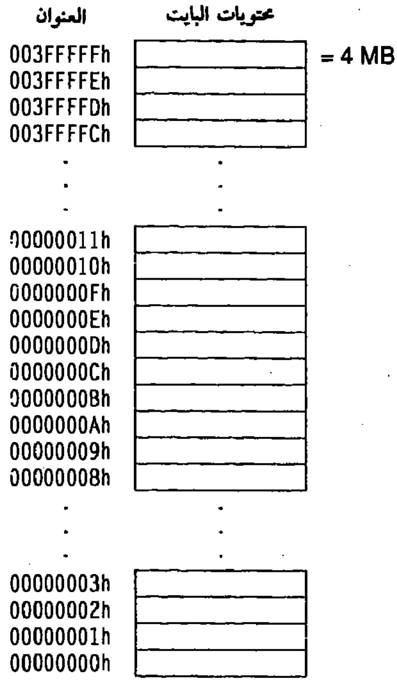
- جعله نقّالاً قدَر الإمكان.
- جعله يعمل بشكل إعتمادي وكافي على كل أحجام التطبيقات دون الحاجة لضبط النظام من قبَل المستعمل أو المدير.
- توفير مزايا إدارة ذاكرة حديثة، مثل الملفّات المخطّطة وذاكرة النسخ على الكتابة ودعم التطبيقات بإستعمال فسحات عنوان كبيرة متفرّقة.
- الإتاحة للمعالجات تحديد موقع الذاكرة الخاصة وإدارتها.
- توفير آليات تدعم الأنظمة الفرعيّة للمحيط مثل الإتاحة لنظام فرعي (ذات حقوق وصول مناسبة) إدارة الذاكرة الظاهريّة لمعالجة مستضاف.
- موازنة حاجات المعالجة المتعدّدة مع سرعة الوصول إلى الذاكرة. (مثلاً، تستطيع حماية بنيات البيانات بإستعمال مستويات متعدّدة من القفل زيادة التوازي في برنامج إدارة VM لكن كل قفل يؤدّي إلى كلفة إضافيّة).

يبدأ القسم التالي بمقدمة عن أنظمة الذاكرة الظاهريّة. بعد ذلك، تعرض معالجة لنموذج الذاكرة الظاهريّة في NT ـ برنامج إدارة VM ـ والمزايا الإضافيّة والخدمات التي يوفّرها برنامج إدارة VM للأنظمة الفرعيّة للمحيط. يتبع ذلك وصف إستخدام برنامج إدارة VM بما في ذلك شمل بنيات البيانات الأساسيّة واللورغاريتم.

1-6 الذاكرة الظاهريّة:

تتّصف الذاكرة بعدّة خصائص: بنية فعليّة وبنية منطقيّة والطريقة التي يترجم بواسطتها نظام التشغيل (أو عدم ترجمته) من بنية واحدة إلى أخرى.

تنظّم الذاكرة الفعليّة كسلسلة من وحدات تخزين من 1 بايت. وترقّم البايتات بدءاً من 0 وتمتدّ إلى كميّة الذاكرة المتوفّرة في تشكيل النظام (ناقص 1) كما يبينّ في الشكل (6-1). تتألّف مجموعة الأرقام هذه (المبيّنة هنا بنسق ست عشري) من فسحة العنوان الفعلي للماكنة.



الشكل (1-6)

فسحة العنوان الفعلي

الذاكرة المنطقيّة والتي تسمّى عادة الذاكرة الظاهريّة، هي طريقة مشاهدة برنامج للذاكرة، وهي نادراً ما تتوافق مع بنية الذاكرة الفعليّة في أنظمة التشغيل الحديثة. تعتمد عادة أنظمة الذاكرة الظاهريّة إما معاينة مقطعيّة أو معاينة خطيّة للذاكرة. ولقد إستعملت كل الحواسيب الشخصيّة المعتمدة على شرائح Intel، من Intel 8086 إلى 80286، نموذج مقطعي. يقسم نظام العنونة المقطعي الذاكرة الفعليّة إلى وحدات من عناوين متّصلة تسمى قطع. يشمل العنوان النموذجي رقم القطعة والحيد ضمن القطعة.

من الناحية المقابلة، تدعم معظم معالجات RISC وحتى معالجات CISC الحاليّة من Intel، ببنية عنونة خطيّة. تتجانس العنونة الخطيّة مع بنية الذاكرة الفعليّة أكثر من العنونة المقطعيّة. تبدأ العناوين في مخطّط خطي عند O وتمتدّ بايت بعد بايت إلى الحدود العليا لفسحة العنوان.

فسحة العنوان الظاهري هي مجموعة عناوين الذاكرة المتوفّرة للإستعمال من قبل شعَب اللولبة. تتّصف كل معالجة بفسحة عنوان ظاهري فريدة أي أنها عادة أكبر من الذاكرة الفعليّة. ورغم أن عدد العناوين الفعليّة على حاسوب معيّن محدود بكميّة الذاكرة المتوفّرة في الحاسوب

(مع تحديد عنوان فريد لكل بايت)، يحد عدد العناوين الظاهريّة فقط بعدد البتّات في عنوان ظاهري. ويمكن لكل بت أن يكون فعّالاً أو ملغى التفعيل. وهكذا، وعلى سبيل المثال، فإن المعالج MIPS R4000 الذي يحتوي عناوين من 32 بت، يحتوي على فسحة عنوان ظاهري من 2^{32} أو أربع مليارات بايت (4 جيجابايت)، كما يوضّح الشكل (6-2).

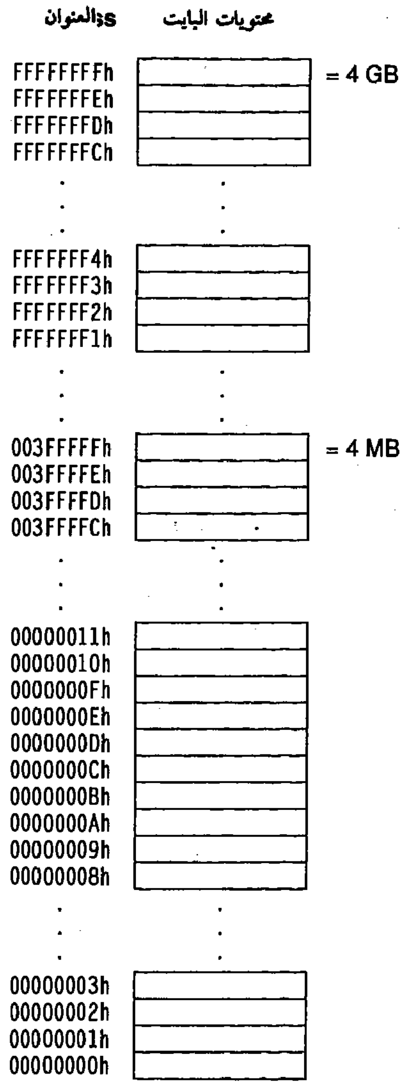
يفرض الفرق بين فسحة عنوان فعلي وفسحة العنوان الظاهري مهمّتي نظام الذاكرة الظاهريّة:

- لترجمة أو تخطيط مجموعة فرعيّة لكل عنوان ظاهري للمعالجة إلى مواقع ذاكرة فعليّة. فعندما تقرأ شعبة أو تكتب فسحة العنوان الظاهري الخاصة بها، يستعمل نظام الذاكرة الفعلي (بعضها يستخدم في العتاد) العنوان الظاهري لإيجاد العنوان الفعلي الصحيح قبل نقل البيانات.
- لمبادلة بعض محتويات الذاكرة إلى قرص عندما تمتلىء الذاكرة ـــ أي، عندما تحاول الشعَب في النظام إستعمال ذاكرة أكثر مما هو متوفّر فعلياً.

تتيح المهمة الأولى، تخطيط العناوين الظاهريّة إلى عناوين فعليّة، إعادة تحديد موقع برنامج بسهولة في الذاكرة خلال تنفيذه. ينقل نظام الذاكرة الظاهريّة أجزاء من البرنامج إلى قرص ثم إلى الذاكرة، حيث تحدّد موقعها في مكان مختلف. بعد ذلك، تحدّث خرائط الذاكرة المنطقيّة ـــ إلى ـــ الفعليّة للإشارة إلى الموقع الجديد.

تنتج المهمة الثانية، مبادلة محتويات الذاكرة إلى قرص، من المهمة الأولى. ومن الواضح أنه من المستحيل لمعالجة عنونة 4 جيجابايت من الذاكرة عند تواجد 4 ميغابايت من الذاكرة الفعليّة على الماكنة. تحقّق أنظمة الذاكرة الظاهريّة ذلك باستعمال سوّاقة القرص «كذاكرة» مساندة (تسمى مخزن مساند). وعندما تمتلىء الذاكرة الفعليّة، ينتقي نظام الذاكرة الفعليّة البيانات المخزّنة في الذاكرة لإزالتها ثم نقلها مؤقتاً إلى ملفّ على قرص. وعند الحاجة مجدداً للبيانات من قبل شعبة تنفيذيّة، ينقلها نظام الذاكرة الظاهريّة إلى الذاكرة.

ولو كانت عملية نقل البيانات وإرجاعها بين الذاكرة والقرص، التي ينفّذها برنامج إدارة الذاكرة الظاهريّة، تتمّ بمعدّل بت واحد في كل مرة، لكانت عمليّة بطيئة جداً وغير مقبولة. لذلك، قسّمت فسحة العنوان الظاهري إلى كتل متساوية الحجم تسمى صفحات. وبشكل مشابه، تقسم الذاكرة الفعليّة إلى كتل تسمى أُطر الصفحة المستعملة لتثبيت الصفحات. تحتوي كل معالجة على مجموعة من الصفحات من فسحة العنوان الظاهري الخاصة بها المتواجدة في



الشكل (2-6)
فسحة العنوان الظاهري الخطي

الذاكرة الفعليّة في أي وقت. تسمى الصفحات الموجودة في الذاكرة الفعليّة والمتوفّرة فورياً، الصفحات الصالحة. وتسمى الصفحات المخزّنة على قرص (أو الموجودة في الذاكرة لكنها غير متوفّرة فورياً)، الصفحات غير الصالحة، كما يوضح ذلك الشكل (3-6).

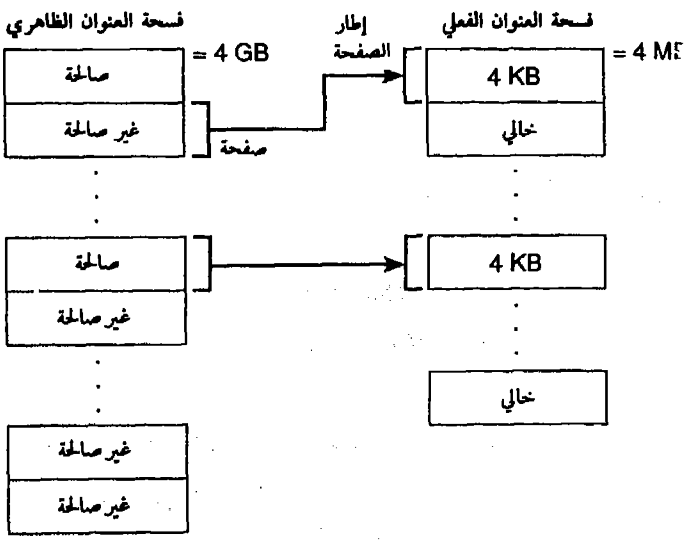
عندما تتمكّن شعبة تنفيذيّة من الوصول إلى ذاكرة ظاهريّة في صفحة معلّمة «غير صالحة»

يصدر المعالج مصيدة نظام تسمى خطأ صفحة. يحدّد نظام الذاكرة الظاهريّة موقع الصفحة المطلوبة على القرص وتحملها في إطار صفحة خالية في الذاكرة الفعليّة. وعندما ينخفض عدد أطر الصفحات المتوفّرة، ينتقي نظام الذاكرة الظاهريّة أُطُر صفحات لإخلاء ونسخ محتوياتها إلى قرص. وهذه العمليّة، المعروفة بإسم الترتيب بصفحات، غير مدركة للمبرمج.

يمكن أن تكون أخطاء الصفحة عمليات مكلفة تتطلّب عدّة دورات معالج لإتمامها. لكن أحجام الصفحة الكبيرة تحيد هذه الكلفة لأنه يحمّل بيانات أكثر في الذاكرة لكل خطأ صفحة وبالتالي يحصل عدد أقل من أخطاء الصفحة. (بالطبع، قد يسبّب حجم الصفحة الكبيرة إلى تحميل أكثر مما يلزم من البيانات، لذلك يجب تحقيق توازن بين أحجام الصفحات الكبيرة والصغيرة). إن عدد البايت في صفحة هو أس من 2 وهو يحدّد عادة من قبل العتاد. يستعمل النظام Windows NT حجم الصفحة المحقّق من قبل Intel 386 والذي هو 2^{12} أو 4 كيلوبايت. (يتيح المعالج MIPS R4000 للبراجيات تحديد حجم الصفحة).

رغم أن تخطيط العناوين الظاهريّة إلى عناوين فعليّة ونقل البيانات إلى مخزن المساندة ومنه هي المهام الأساسيّة لنظام الذاكرة الظاهريّة، يجب عليه تنفيذ مهام أخرى عديدة أيضاً:

- يجب أن يتيح لمعالجتين مشاركة الذاكرة بسهولة وفعاليّة.
- يجب أن يحمي الذاكرة المشاركة والخاصة من الوصول غير المسموح به.



الشكل (3-6)
تخطيط أطر الصفحات الظاهريّة إلى صفحة فعليّة

- إذا كان يعمل على حواسيب متعدّدة المعالجات، كما يعمل النظام Windows NT، يجب أن يجيب على أخطاء الصفحة من أكثر من شعبة واحدة في كل مرة.

إن طريقة تنفيذ نظام الذاكرة الظاهريّة للبرنامج التنفيذي NT، برنامج إدارة VM هي موضوع بقيّة هذا الفصل.

2-6 مزايا نمط المستعمل:

يوفّر برنامج إدارة NT VM وظائفيّة عينيّة للمعالجات في نمط المستعمل عبر خدماته المحليّة. تستعمل الأنظمة الفرعيّة للمحيط الخدمات لإدارة معالجات المستضاف العائدة لها. كذلك يصدّر النظام الفرعي WIN 32 بعض القدرات المتوفّرة من قبل خدمات الذاكرة المحليّة في روتين Win 32 API.

يتيح برنامج إدارة VM للأنظمة الفرعيّة في نمط المستعمل مشاركة الذاكرة بفعاليّة باستعمال الكائنات المحميّة والمسمّاة والمناولة مثل الكائنات التنفيذيّة الأخرى. تستطيع الأنظمة الفرعيّة ضبط الحماية لمستوى الصفحة على الذاكرة الخاصة، ويمكنها قفل الصفحات المنتقاة في الذاكرة، ويمكنها أن تستعمل الملفّات المخطّطة وإدارة فسحات العنوان الظاهري لمستضافاتها.

تركّز الأقسام الفرعيّة التالية على قدرات جعل برنامج إدارة VM متوفّراً لنمط المستعمل، إدارة فسحة عنوان ظاهري لمعالجة، ومشاركة الذاكرة بين المعالجات وحماية ذاكرة ظاهريّة لمعالجة واحدة من المعالجات الأخرى.

1-2-6 إدارة الذاكرة:

كما أظهر الفصل الرابع «المعالجات والشعَب» في مخططات الصفحات والخدمات لكائنات المعالجة (الشكل (4-3))، يزوّد برنامج إدارة VM مجموعة من الخدمات المحليّة تستطيع المعالجة إستعمالها لإدارة ذاكرتها الظاهريّة مباشرة. تتيح هذه الخدمات لمعالجة تنفيذ ما يلي:

- تحديد موقع الذاكرة في معالجة من مرحلتين.
- قراءة الذاكرة الظاهريّة وكتابتها.
- قفل الصفحات الظاهريّة في الذاكرة الفعليّة.
- إحضار المعلومات المتعلّقة بالصفحات الظاهريّة.
- حماية الصفحات الظاهريّة.
- نقل الصفحات الظاهريّة إلى قرص.

ينشىء برنامج إدارة VM طريقة بمرحلتين لتحديد موقع الذاكرة ـ حجزها ثم إعتمادها. الذاكرة المحجوزة هي مجموعة من العناوين الظاهريّة التي حجزها برنامج إدارة VM لإستعمال المعالجة المستقبلي. إن حجز الذاكرة (أي العناوين الظاهريّة) هي عمليّة سريعة ورخيصة في Windows NT . الذاكرة المعتمدة هي ذاكرة حدّد لها برنامج إدارة VM فسحة في ملفّ الصفحات وهو ملفّ القرص حيث تكتب الصفحات الظاهريّة عند إزالتها من الذاكرة. وعندما تحدّد شعبة موقع الذاكرة الظاهريّة فإنها تستطيع حجز الذاكرة واعتمادها في نفس الوقت أو يمكنها حجز الذاكرة واعتمادها فقط عند الضرورة.

إن حجز الذاكرة مفيد عندما تنشىء الشعبة بنيات بيانات ديناميّة. تحجز الشعبة تتابعاً من العناوين الظاهريّة التي تعتمدها عند الضرورة لاحتواء البيانات. وإذا وجب أن تنمو بنية البيانات، يمكن أن تعتمد الشعبة ذاكرة إضافيّة من المنطقة المحجوزة. تضمن هذه الإستراتيجيّة عدم محاولة شعَب أخرى تشتغل ضمن المعالجة (رزمة مكتبة، مثلًا) أو معالجة أخرى (مثل شعبة النظام الفرعي Win 32) إستعمال عناوين ظاهريّة متّصلة قد تحتاجها بنية البيانات للتمدّد.

تستطيع شعبة إنتقاء العنوان الظاهري البادىء لمنطقة محجوزة، أو يمكنها الإتاحة لبرنامج إدارة VM إيجاد مكان لها في فسحة العنوان الظاهري للمعالجة.

يحسم برنامج إدارة VM من حصة ملف صفحات المعالجة للذاكرة المعتمدة لكن ليس للذاكرة المحجوزة. يتيح هذا المستوى المزدوج من الألسنيّة لشعبة حجز منطقة كبيرة من الذاكرة الظاهريّة مع تجنّب محاسبة حصّتها إلى أن تتمّ الحاجة للذاكرة. وهو يساعد على إبقاء ملف الصفحات خالياً لصفحات الذاكرة الظاهريّة المستعملة فعلياً. وعند عدم إستعمال مجال معيّن من العناوين، يمكن لشعبة أن «تلغي إعتمادها» وبالتالي تخلي الفسحة في ملفّ الصفحات وتستعيد حصة ملفّ صفحات المعالجة. (راجع الفصل الرابع «المعالجات والشعَب» لمزيد من المعلومات المتعلّقة بحصص المعالجة).

بالنسبة للتطبيقات الزمنيّة وتلك مع متطلّبات الأداء الأخرى، يتيح برنامج إدارة VM لنظام فرعي في نمط المستعمل أو معالجة أخرى بتفضيلات خاصة، قفل صفحات ظاهريّة محدّدة في الذاكرة. وهذا يضمن أنه لم تتمّ إزالة صفحة حرجة من الذاكرة خلال إشتغال أية شعبة في المعالجة. فمثلاً، قد يختار تطبيق قاعدة بيانات، يستعمل بنية شجرة للمحافظة على بياناته، قفل جذور الشجرة في الذاكرة بحيث لا يؤدّي الوصول إلى قاعدة البيانات إلى أخطاء صفحة غير ضروريّة.

مثل خدمات NT الأخرى، تتيح خدمات VM للمستدعي تزويد مقبض معالجة ليشير إلى

معالجة يجب مناولة ذاكرتها الظاهريّة. يستطيع المستدعي مناولة ذاكرته الظاهريّة أو تلك العائدة لمعالجة أخرى، هذه القدرة قويّة لأنها تتيح معالجة واحدة في نمط المستعمل إدارة فسحة عنوان أخرى. فمثلاً، تستطيع معالجة واحدة إنشاء معالجة أخرى، وتزويدها بحق مناولة الذاكرة الظاهريّة للمعالجة الجديدة. بعد ذلك، تستطيع المعالجة الأولى تحديد موقع الذاكرة وإخلائها وقراءتها وكتابتها نيابة عن المعالجة الثانية باستدعاء خدمات الذاكرة الظاهريّة وتمرير مقبض المعالجة الثانية تستعمل هذه المزيّة من قبَل الأنظمة الفرعيّة لإدارة ذاكرة معالجات مستضافاتها.

تستطيع تطبيقات Win 32 الوصول إلى العديد من قدرات إدارة VM هذه عبر روتين Win 32 API. ويمكنها تحديد موقع الذاكرة الظاهريّة وإخلائها وقراءة الذاكرة الظاهريّة وكتابتها ونقل الصفحات الظاهريّة إلى قرص وإحضار المعلومات المتعلّقة بمجال صفحات ظاهريّة وقفل الصفحات الظاهريّة في الذاكرة وحماية الصفحات المحدّدة. لا يتيح أي من روتينات API هذه لبرنامج Win 32 الإستيلاء على الذاكرة الظاهريّة لمعالجة أخرى باستثناء روتين ReadProcessMemory () وروتين WriteProcessMemory (). وهذه مخصّصة للإستعمال من قبل مزيلات العِلل في نمط المستعمل لإنشاء نقاط فصل والمحافظة على بيانات الحالة الآنيّة لمعالجة قيد إزالة العلل.

2-2-6 مشاركة الذاكرة:

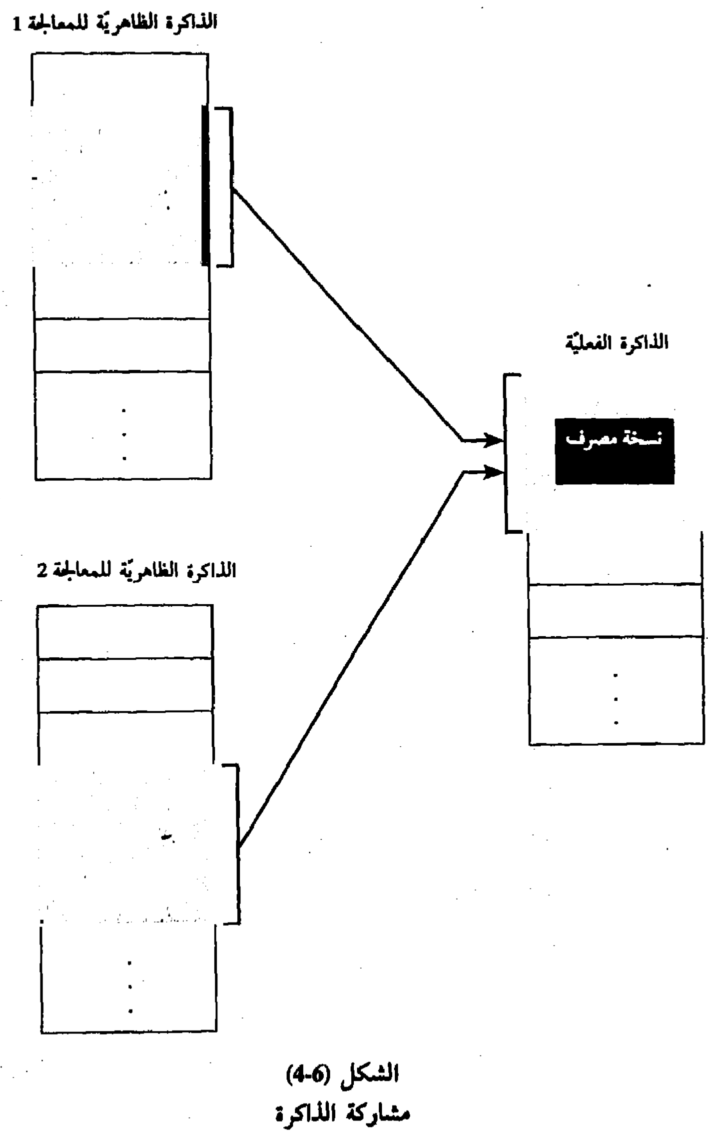
إحدى المهام المهمّة لنظام إدارة الذاكرة هي الإتاحة للمعالجات مشاركة الذاكرة عندما تحتاجها أو عندما تجعل المشاركة من نظام التشغيل أكثر كفاية. فمثلاً، إذا كانت معالجتان تصرّفان برامج C، يمكن تخفيض إستعمال الذاكرة إذا تمّ تحميل نسخة واحدة من المصرّف C في الذاكرة (وطبعاً، يجب أن تحتجز كل معالجة مناصف ذاكرة خاصة حيث تسجّل الشيفرات والبيانات الخاصة).

توفّر الذاكرة الظاهريّة آلية مناسبة لمشاركة الذاكرة. ولأن كل معالجة تحتوي فسحة عنوان ظاهري مستقلّة، يستطيع نظام التشغيل تحميل المصرّف في الذاكرة مرة واحدة وعندما تحفز المعالجة المصرّف، يستطيع برنامج إدارة VM تخطيط العناوين الظاهريّة للمعالجة الثانية إلى أُطُر الصفحة الفعليّة المشغولة من قبل المصرّف، كما يوضّح ذلك الشكل (6-4).

وبشكل مشابه، إذا أنشأت معالجتان متعاونتان مخزناً مؤقّتاً لذاكرة مشاركة، يمكن تخطيط فسحة العنوان الظاهري لكل منها إلى نفس أُطُر الصفحة الفعليّة المشغولة من قبَل المخزن المؤقت. في مثال المصرّف، لا يتيح برنامج إدارة VM لأية معالجة تعديل الصفحات المشغولة من قبل المصرّف. وقد حدّد الصفحات الظاهريّة في المعالجتين على أنها مقروءة فقط. لكن في مثال

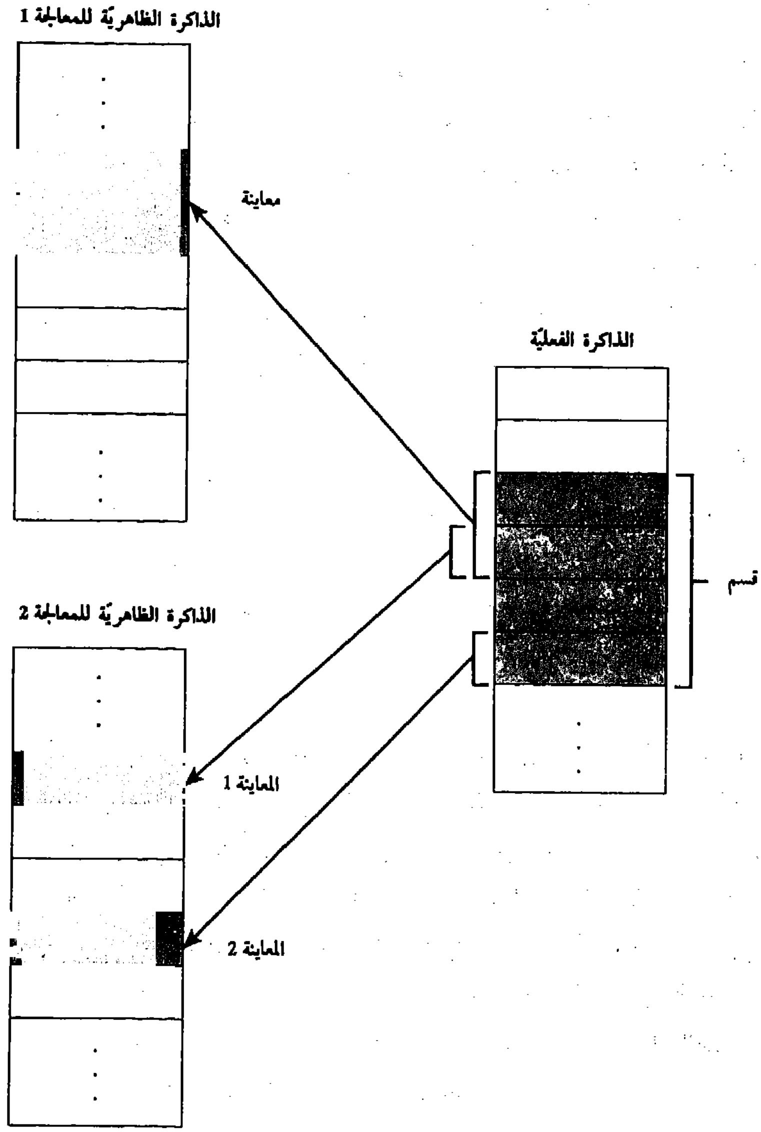
المخزن المؤقّت، قد تحتاج الشعَب في المعالجتين للكتابة إلى المخزن المؤقّت المشارك. لذلك، حدّدت الصفحات على أنها قراءة / كتابة. وبالطبع وعند مشاركة بنية بيانات بهذه الطريقة، يجب على الشعَب التي تستعملها مزامنة وصولها إلى الذاكرة المشاركة لمنع الوصول في نفس الوقت وتخريب البيانات. (يتمّ وصف حماية الذاكرة في القسم 6-2-3).

يوفّر النظام الفرعي Win 32 قدرات مشاركة الذاكرة في البرنامج التنفيذي NT لتطبيقات



الشكل (6-4)
مشاركة الذاكرة

Win 32 بواسطة روتينات API لتخطيط الملفّ الخاص بها وهو موضوع سيُشرح في القسم الفرعي التالي.



الشكل (6-5)
تخطيط معاينات قسم

1-2-2-6 الأقسام والمعاينات والملفّات المخطّطة:

مثل سائر مكوّنات Windows NT، يتوازى بالكامل برنامج إدارة VM. وهو يشتغل في نفس الوقت على كل المعالجات في حاسوب متعدّد المعالجات ويجب أن يشارك بنيات بياناته ضمن الشعَب العاملة على معالجات مختلفة. لذلك، كان من المهمّ إنشاء حلّ آمن وكافي لمشاركة الذاكرة في Windows NT، وليس فقط لبرامج نمط المستعمل لكن أيضاً للنظام نفسه.

يمكن تعريف الذاكرة المشاركة على أنها ذاكرة مرئيّة من أكثر من معالج واحد أو أنها موجودة في أكثر من فسحة عنوان ظاهري واحد. إن طريقة Windows NT لمشاركة الموارد هي في إستخدامها ككائنات محميّة والذاكرة ليست إستثناء. يمثّل كائن القسم الذي يوفّره النظام الفرعي Win 32 ككائن تخطيط ملفّ، كتلة من الذاكرة تستطيع معالجتان أو أكثر مشاركتها. تنشىء شعبة في معالجة واحدة كائن قسم ويحدّد له إسم لكي تتمكّن الشعَب في المعالجات الأخرى فتح مقابض إليه. بعد فتح مقبض إلى كائن قسم، تستطيع الشعبة تخطيط القسم أو أجزاء من القسم في فسحة العنوان الظاهري الخاص بها (أو معالجة أخرى).

يمكن أن يكون كائن قسم NT كبيراً من عشرات أو مئات أو حتى آلاف الصفحات. ولحفظ فسحة العنوان الظاهري الخاصة بها، يجب على المعالجة أن تخطّط جزءًا من كائن القسم المطلوب. يسمى الجزء الذي تخطّطه معاينة قسم. توفّر المعاينة إطار إلى منطقة الذاكرة المشاركة وتستطيع المعالجات المختلفة تخطيط معاينات مختلفة أو حتى متعدّدة لقسم، كما يبينّ في الشكل (6-5).

يتيح تخطيط معاينات قسم لمعالجة الوصول إلى كتل كبيرة من الذاكرة التي لا تحتوي في حالات أخرى على فسحة عنوان ظاهري كافية للتخطيط. مثلاً، قد تحتوي شركة على قاعدة بيانات كبيرة تحتوي معلومات تتعلّق بموظفيها. ينشىء برنامج قاعدة البيانات كائن قسم ليحتوي على قاعدة بيانات الموظفين بأكملها. وعندما يستعلم مستعمل قاعدة البيانات، يخطّط البرنامج معاينة قسم قاعدة البيانات إلى فسحة عنوانها الظاهري ويحضر المعلومات منها ويزيل تخطيط المعاينة ثم يخطّط معاينة أخرى للقسم لإحضار مزيد من المعلومات. في الواقع، يستعرض البرنامج كائن القسم الكبير هذا منطقة واحدة في كل مرة حيث يحصل على البيانات من كل جزء من قاعدة البيانات دون النفاذ من فسحة العنوان الظاهري.

ومثل الذاكرة الخاصة، ترتب محتويات الذاكرة المشاركة في صفحات إلى قرص عندما يكون الضغط على الذاكرة كبيراً. يكتب برنامج إدارة VM معظم الصفحات، الخاصة

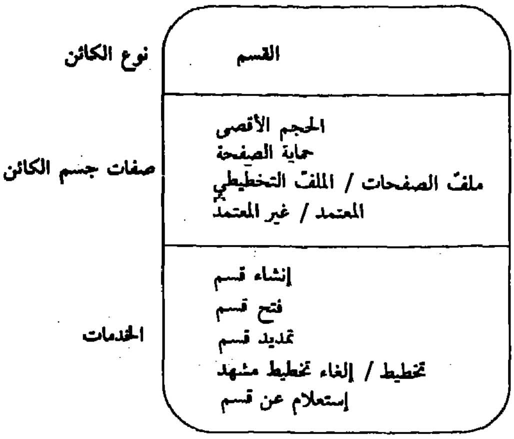
والمشاركة، إلى ملف الصفحات عندما يزيلها من الذاكرة. لكن برنامج إدارة VM يتيح أيضاً وضع كائنات القسم في صفحات إلى ملفّ تخطيطي. وقاعدة بيانات موظفي الشركة هي مثال عن الملفّ التخطيطي. يستعمل برنامج قاعدة البيانات كائن القسم لجلب محتويات ملفّ قاعدة البيانات إلى الذاكرة الظاهريّة. يستطيع البرنامج الوصول إلى الملفّ كصفيفة كبيرة بواسطة تخطيط معاينات مختلفة من كائن القسم وقراءة الذاكرة أو الكتابة إليها عِوَضاً عن الملفّ (وهذا ما يسمى بدخل / خرج الملفّ التخطيطي). وعند وصول البرنامج إلى صفحة غير صالحة (صفحة غير موجودة في الذاكرة الفعليّة)، يحصل خطأ صفحة، ويجلب برنامج إدارة VM تلقائياً الصفحة إلى الذاكرة من الملفّ التخطيطي. وإذا عدّل التطبيق الصفحة، يكتب برنامج إدارة VM التغييرات إلى الملفّ خلال عمليات ترتيب الصفحات العاديّة.

يستعمل البرنامج التنفيذي NT الملفّات التخطيطيّة لتحميل الرسوم القابلة للتنفيذ إلى الذاكرة ويستعمل برنامج إدارة مخبأ النظام الملفّات التخطيطيّة لقراءة الصفحات المخبّأة والكتابة إليها. يستعمل نظام الدخل / الخرج في NT الملفّات التخطيطيّة للذاكرة لتنفيذ طلبات الدخل / الخرج والإتاحة لبرنامج إدارة VM وضع أي تغيير في صفحات إلى القرص كجزء من عمليات ترتيب الصفحات العادية.

تستطيع تطبيقات Win 32 إستعمال الملفّات التخطيطيّة لتنفّذ بشكل مناسب الدخل / الخرج العشوائي (إضافة إلى الدخل / الخرج التتابعي) على ملفّات كبيرة. ينشىء التطبيق كائن تخطيط ملفّ Win 32 (الذي يتوافق مع كائن قسم NT) ليحتوي الملفّ ثم يقرأ أو يكتب إلى مواقع عشوائيّة في الملفّ. يضع برنامج إدارة VM الأجزاء الضروريّة من الملفّ في صفحات تلقائياً ويكتب أي تغييرات على القرص.

2-2-2-6 كائن القسم:

تحدّد مواقع كائنات القسم، كسائر الكائنات الأخرى، ويلغى موقعها بواسطة برنامج إدارة الكائنات. ينشىء برنامج إدارة الكائنات ترويسة كائن ويحفزها والتي يستعملها وِدارة الكائنات. يعرف برنامج إدارة VM جسم كائن القسم. كذلك يستخدم برنامج إدارة VM الخدمات التي تستطيع الشعَب في نمط إستدعاءها لإسترداد الصفات المخزّنة في جسم كائنات القسم وتغييرها. يبينّ كائن القسم في الشكل (6-6).



الشكل (6-6)
كائن القسم

يلخِّص الجدول التالي الصفحات الفريدة المخزَّنة في كائنات القسم:

يؤدي تخطيط مشهد كائن قسم إلى إظهار جزء من القسم في فسحة العنوان الظاهري للمعالجة. وبشكل مشابه، يؤدي إلغاء تخطيط مشهد قسم إلى إزالته من فسحة العنوان الظاهري للمعالجة.

الجدول (6-1)
صفات كائن القسم

الصف	الغرض
الحجم الأقصى	أقصى حجم يمكن أن يبلغه القسم بالبايت. في حال تخطيط ملفّ يكون بحجم الملفّ.
حماية الصفحة	حماية الذاكرة المعتمدة على الصفحة المعيّنة لكل الصفحات في القسم عند إنشائها.
ملفّ الصفحات/الملفّ التخطيطي	يشير إلى أن القسم منشأ فارغ (مدعوم بملفّ صفحات) أو محمّل بملفّ (مدعوم بالملفّ التخطيطي).
المعتمد/غير المعتمد	تشير إلى كون القسم قسماً معتمداً والذي يجب أن يظهر عند نفس العنوان الظاهري لكل المعالجات التي تشاركه، أو قسم غير معتمد الذي يمكن أن يظهر عند عناوين ظاهريّة مختلفة لمعالجات مختلفة.

تحصل المشاركة عندما تخطّط معالجتان أجزاءً من نفس كائن القسمٍ في فسحة عناوينها. وعندما تشارك معالجتان بهذه الطريقة، يجب أن تزامنا وصولهما إليه لتجنّب تغيير البيانات في نفس الوقت. يمكن إستعمال الأحداث والإعلام الإستشاري أو حتى الأقفال المعتمدة على العتاد

لمزامنة الوصول إلى قسم مشارك. لا تعرّف كائنات القسم على أنها كائنات مزامنة، أي، لا تستطيع شعبة مزامنة تنفيذها بانتظار مقبض إلى كائن قسم. تستطيع تطبيقات Win 32 إستعمال الخوافت والأحداث والأقسام الحَرِجة أو الأعلام الإستشاري لمزامنة وصولها إلى كائن تخطيط الملفّ ـ ما يعادلها في كائن قسم.

لتخطيط مشهد قسم، يجب على المعالجة أن تطلب مقبضاً لها بعد ذلك، تحتوي المعالجة التي تنشىء كائن القسم على مقبض دائماً. تستطيع المعالجات الأخرى (تلك ذات حقوق الوصول المناسبة) فتح مقابض إلى كائن القسم يحتوي على إسم. وبشكل بديل، يمكن تعيين مقبض معالجة إلى كائن قسم عبر تأصيل المعالجة أو عندما تستنسخ المعالجة الأخرى مقبض القسم العائد لها وتمرّر المقبض المستنسخ إلى المعالجة المستسلمة. تتمّ مشاركة الذاكرة في كل هذه الحالات. وإذا تمّ إنشاء قسم مشارك ككائن مؤقّت، يحذف برنامج إدارة الكائنات الذاكرة المشاركة عند إفلات المرجع الأخير إلى كائن القسم. ولا تحذف كائنات القسم الدائمة.

3-2-6 حماية الذاكرة:

تتوفّر حماية الذاكرة في Windows NT في أربعة نماذج، الثلاثة الأوّل عامّة لمعظم أنظمة التشغيل الحديثة:

- فسحة عنوان مستقلّ لكل معالجة، لا تتيح العتاد لأية شعبة الوصول إلى العناوين الظاهريّة لمعالجة أخرى.
- نمَطًا تشغيل: نمط النواة الذي يتيح للشعَب الوصول إلى شيفرة النظام وبياناته؛ ونمط المستعمل الذي لا يتيح ذلك.
- آلية حماية تعتمد على الصفحة. تحتوي كل صفحة ظاهريّة على مجموعة من الأعلام المتعلّقة بها تحدّد أنواع الوصول المتاح في نمط المستعمل وفي نمط النواة.

وتوفّر الآلية التالية، الفريدة للنظام Windows NT، نموذجاً واحداً إضافياً لحماية الذاكرة:

- حماية الذاكرة المعتمدة على كائن. في كل مرّة تفتح معالجة مقبض إلى كائن قسم أو تخطّط معاينة له، يدقّق مراقب مراجع الأمان في Windows NT لجهة الإتاحة للمعالجة الوصول إلى الكائن.

تركّز الأقسام الفرعيّة التالية على نوعين من حماية الذاكرة التي تدعمها هذه الآليات ــ حماية الذاكرة الخاصة بالمعالجة وحماية الذاكرة المشاركة.

1-3-2-6 الذاكرة الخاصة بالمعالجة:

في كل مرة تستعمل الشعبة عنواناً، يتدخّل برنامج إدارة VM للبرنامج التنفيذي NT سويّة مع العتاد، ويترجم العنوان الظاهري إلى عنوان فعلي. يستطيع نظام الذاكرة الظاهريّة، عن طريق التحكُّم بترجمة العناوين الظاهريّة، ضمان عدم وصول الشعَب في معالجة واحدة إلى إطار صفحة ذاكرة يعود إلى معالجة أخرى.

إضافة إلى الحماية الخاصة المتوفّرة من قبل ترجمة العنوان الظاهري ــ إلى ــ فعلي، يوفّر كل معالج بدعم ذاكرة ظاهريّة لنموذج ما لحماية الذاكرة المحكومة بالعتاد. لكن الحمايات التي توفّرها واستخدامات العتاد تتغيّر. وتكون حماية العتاد في غالب الأحيان، منخفضة جداً ويجب تزويدها بآليات مزوّدة من قبل برامجيات الذاكرة الظاهريّة. وهذا الواقع يجعل برنامج إدارة VM في Windows NT عرضة لإختلافات العتاد بالمقارنة مع أي جزء آخر من نظام التشغيل.

تنفّذ حماية الصفحة المعتمدة على العتاد في كل مرّة تقوم فيها الشعبة بالوصول إلى الذاكرة. فمثلاً، على المعالج MIPS R4000، تحدّد كل صفحة ذاكرة ظاهريّة لمعالجة إما كصفحة في نمط المستعمل (2 جيغابايت منخفض) أو صفحة في نمط النواة (2 جيغابايت مرتفع) وأما صفحة مقروءة فقط أو قراءة / كتابة. إذا كانت الشعبة تشتغل في نمط النواة، يتيح المعالج لها قراءة أية صفحة صالحة من الذاكرة وكتابة الصفحات الصالحة بالتحديد قراءة / كتابة. وإذا كانت الشعبة تشتغل في نمط النواة، يمكنها فقط قراءة صفحات المستعمل الصالحة. ويستطيع كتابة فقط صفحات المستعمل الصالحة ذات التحديد قراءة / كتابة. يصدر المعالج MIPS R4000 خطأ صفحة إذا كانت الصفحة التي تمّ الوصول إليها غير صالحة (ليست في الذاكرة). وهو يصدر إستثناء خطأ عنوان (مخالفة وصول) إذا حاولت الشعبة قراءة أو كتابة صفحة صالحة بطريقة مخالفة للقواعد.

تستطيع العتاد تنفيذ تدقيقات الحماية الخاصة بها فقط على الصفحات الصالحة ــ تلك المتواجدة في الذاكرة. وإذا استطاعت شعبة الوصول إلى صفحة غير صالحة (واحدة غير موجودة في الذاكرة)، يصدر المعالج MIPS R4000 خطأ صفحة، وتتولّى برامجيات صفحات برامج إدارة VM مهمّة حماية الصفحة.

يوفّر برنامج إدارة VM نفس حمايات الصفحة التي يزوّدها المعالج MIPS 4000 للصفحات الصالحة:

- قراءة فقط.
- قراءة / كتابة.

يزوّد برنامج إدارة VM هذه الحمايات الأساسيّة مع أخرى خاصة به:

- التنفيذ فقط (إذا كانت العتاد تدعمه).
- صفحة الوقاية.
- اللاوصول.
- النسخ عند الكتابة.

باستعمال خدمات الذاكرة الظاهريّة المحليّة، يستطيع النظام الفرعي لمحيط التحكُّم بحماية مستوى الصفحة على صفحات ظاهريّة خاصة. يؤدّي التحكُّم بحماية مستوى الصفحة إلى برامج أكثر إعتماديّة عن طريق ضمان عدم كتابة الشعَب إلى صفحات التي تُقرأ فقط. تفيد هذه القدرة أيضاً، على إزالة العِلل من برنامج متعدّد الشعَب حيث تكتب شعبة واحدة خطأً إلى الذاكرة. وعن طريق التغيير المؤقت لحماية الصفحة للقراءة فقط أو لا وصول، يستطيع مزيل العلل إلتقاط الشعبة وإيجاد الخطأ.

لا تستطيع الشعبة القراءة من أو الكتابة إلى الصفحة بواسطة الوصول فقط للتنفيذ لكنها تستطيع القفز إلى عنوان ضمن الصفحة والبدء بالتنفيذ. يناسب هذا النوع من الحماية لبراجيات التطبيقات المشاركة مثل محرّر أو مصرّف، يجب أن تتمكّن كل الشعَب من تشغيل البراجيّات لكن لا يجب أن يُتاح لأية منها القراءة من الرسم المنفّذ أو الكتابة إليه. (لاحظ أن المعالجات MIPS R4000 و Intel 386, 486 لا تدعم الحماية فقط للتنفيذ. لذلك، وعلى هذه المعالجات، يعادل الوصول للتنفيذ الوصول للقراءة فقط).

يوفّر برنامج إدارة VM حماية صفحة الوقاية لتسهيل تـدقيق الروابط التلقـائيّة عـلى التكديسات، لكن يمكن إستعمال هذا النوع من حماية الصفحة لتحديد بنيات البيانات أيضاً. وعندما تتمكّن شعبة من الوصول إلى صفحة وقاية، ينشىء برنامج إدارة VM إستثناء صفحة وقاية ويستلم المستدعي رسالة راجعتها صفحة الوقاية. بعد ذلك يتيح برنامج إدارة VM متابعة التشغيل. وإذا وضع نظام فرعي أو تطبيق محلي آخر صفحة وقاية في نهاية صفيفة ديناميّة، على سبيل المثال، يستلم النظام الفرعي تحذيراً من برنامج إدارة VM عندما يصل إلى صفحة الوقاية ويمكنه تمديد الصفيفة ديناميا.

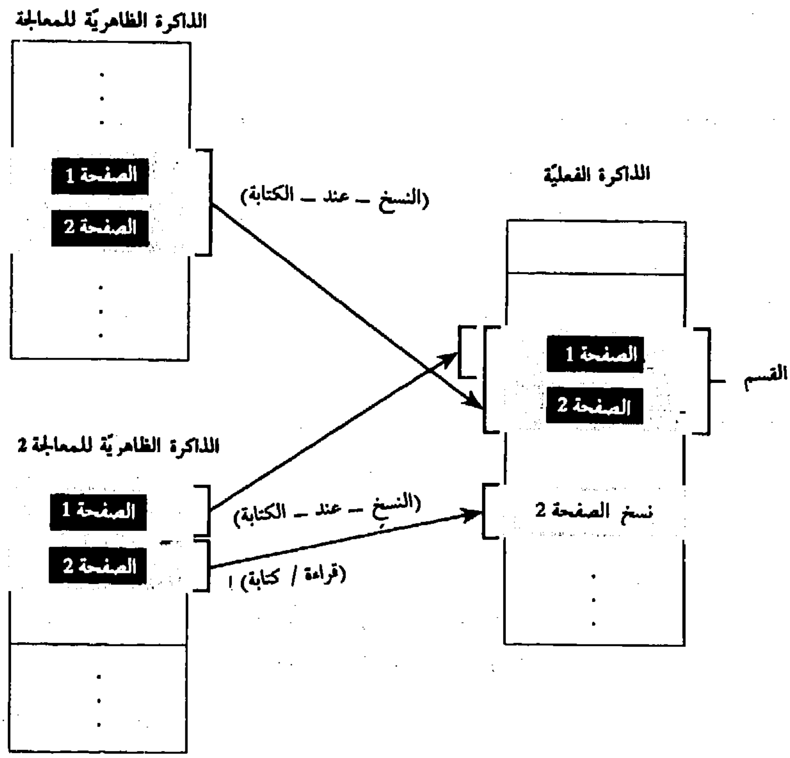
تستعمل حماية صفحة اللا وصول لمنع أية شعبة من قراءة صفحة معيّنة أو الكتابة إليها. يصدر برنامج إدارة VM إستثناءاً إذا تمّ الوصول إلى عنوان في الصفحة. يعينّ للصفحات

الظاهريّة التي لم يتمّ تحديد موقعها أو تلك التي حجزت دون إعتمادها، حماية صفحة اللا وصول من قبل برنامج إدارة VM. تستعمل حماية صفحة اللا وصول أساساً من قبَل مزيلي العلل.

يجعل النظام الفرعي Win 32 حماية صفحة برنامج إدارة VM مرئيّة لتطبيقات Win 32 بواسطة الروتين VirtualProtect (). يتيح هذا الروتين للتطبيقات تحديد الصفحات الظاهريّة الإفراديّة كقراءة فقط وقراءة / كتابة أو لا وصول. لا تتوفّر صفحة وقاية وتنفيذ فقط والنسخ على الكتابة.

2-3-2-6 الذاكرة المشاركة:

إن حماية صفحة النسخ عند الكتابة، المذكورة في القسم الفرعي السابق، هي إستمثال يستعمله برنامج إدارة VM لحفظ الذاكرة. وعندما تريد معالجتان قراءة وكتابة نفس محتويات



الشكل (6-7)
حماية النسخ ـ عند ـ الكتابة

الذاكرة (لكن دون مشاركتها). يعيّن برنامج إدارة VM حماية صفحة النسخ عند الكتابة إلى منطقة الذاكرة. ثم يشارك الذاكرة الفعليّة بين المعالجات طالما أن أي منها لا يكتب إلى صفحة. وفي حال كتابة شعبة في إحدى المعالجات إلى صفحة، ينسخ برنامج إدارة VM إطار الصفحة الفعليّة إلى موقع آخر في الذاكرة ويحدّث فسحة العنوان الظاهري للمعالجة للإشارة إلى النسخ ويضبط حماية الصفحة الجديدة إلى قراءة / كتابة. وكما يبيّن في الشكل (6-7)، فإن الصفحة المنسوخة غير مرئيّة إلى الشعَب في المعالجات الأخرى. وهكذا، تستطيع الشعبة الكتابة إلى نسخة الصفحة العائدة لها دون التأثير على المعالجات الأخرى التي تستعمل الصفحة.

تفيد حماية النسخ عند الكتابة للصفحات التي تحتوي شيفرة. فهي تضمن تأثير فقط المعالجة التي تقوم شعبها بتعديل الرسم، بهذا التنفيذ. فمثلاً، تبدأ صفحات الشيفرة كصفحات تنفيذ فقط. لكن، إذا ضبط المبرمج نقاط فصل خلال عمليّة إزالة العلل من برنامج، يجب أن يضيف مزيل العلل تعليمات نقطة فصل إلى الشيفرة. وللقيام بذلك، فهو يغيّر أولاً الحماية على الصفحة إلى النسخ عند الكتابة. ينشيء برنامج إدارة VM فوراً نسخة خاصة لصفحة الشيفرة للمعالجة ذات الشعبة التي ضبطت نقطة الفصل. تتابع المعالجات الأخرى إستعمال الشيفرة غير المعدّلة. لا يوفّر النظام الفرعي Win 32 مباشرة حماية صفحة النسخ عند الكتابة إلى تطبيقات Win 32 لكنه يستعمل بشكل غير مباشر حماية النسخ عند الكتابة لإستخدام بيانات الحالة الآنيّة لكل معالجة في مكتبات الربط الديناميي (DLLs) العائدة لها وفي أي مكان آخر.

إن حماية صفحة النسخ عند الكتابة هو مثال عن تقنيّة الإستمثال المسمّاة التقييم الكسول الذي يستعمله برنامج إدارة VM حيث أمكن. تتجنّب لوغاريتمات التقييم الكسول تنفيذ عملية مكلفة إلى أن يكون هناك ضرورة لها. وإذا لم تطلب العمليّة، عندئذٍ لا يهدر أي وقت عليها. إن النظام الفرعي POSIX هو أحد المكوّنات التي تستغلّ هذا الإستمثال إلى حدٍّ كبير. وعادة، وعندما تستدعي معالجة روتينAPI () Fork لإنشاء معالجة أخرى على النظام POSIX، ينسخ نظام التشغيل فسحة عنوان المعالجة الأولى إلى الثانية ــ وهي عمليّة تستغرق وقتاً. يستدعي التطبيق الجديد روتين API () exec فوراً الذي يعيد تحفيز فسحة العنوان مع برنامج تنفيذي، بحيث يجعل عمليّة النسخ الأولى غير ضروريّة. لكن من الناحية المقابلة، يعلّم لوغاريتم التقييم الكسول لبرنامج إدارة VM صفحات الأم مع التابع. وإذا لم يتمّ معالجة تابع (أو أم) إلى فسحة عنوانها، تستمرّ المعالجتان بمشاركة ولا يتم نسخ أي شيء. وإذا كتبت واحدة منهما، ينسخ برنامج إدارة VM فقط الصفحات التي كُتبت إليها المعالجة عوَضاً عن نسخ فسحة العنوان بأكملها.

تستخدم كل آليات حماية الذاكرة التي وُصفت إلى الآن إما في العتاد أو في براجيات إدارة

الذاكرة المنخفضة المستوى المحفزة في كل مرة تستعمل فيها شعبة عنوان. يوفّر تصميم كائن Windows NT طبقة إضافيّة من الحماية للذاكرة المشاركة بين معالجتين. يحمي النظام الفرعي للأمان كائنات القسم بنفس الطريقة التي يحمي فيها الكائنات التنفيذيّة الأخرى ـ باستعمال لائحة التحكّم بالوصول (ACL). (راجع الفصل الثالث «برنامج إدارة الكائنات وأمان الكائن») تستطيع الشعبة إنشاء كائن قسم ذات لائحة ACL تحدّد المستعملين أو مجموعة المستعملين الذين يستطيعون قراءة القسم وكتابته وجلب المعلومات المتعلّقة به أو تمديد حجمه.

يدقّق مراقب مراجع الأمان بالحماية على كائن قسم عندما تحاول شعبة فتح مقبض إلى قسم لتخطيط معاينة عنه. وإذا لم تتحْ لائحة ACL العمليّة، يرفض برنامج إدارة الكائنات طلب الإستدعاء. وبعد أن تفتح شعبة بنجاح مقبض إلى قسم، فإن أعمالاً تبقى عرضة للحمايات المعتمدة على الصفحة.

تستطيع الشعبة تغيير حماية مستوى الصفحة على الصفحات الظاهريّة في قسم إذا لم يخالف التغيير لائحة ACL على كائن القسم. فمثلاً، يتيح برنامج إدارة VM لشعبة تغيير صفحات قسم قراءة فقط للحصول على وصول إلى النسخ عند الكتابة وعدم الحصول للوصول إلى كتابة / قراءة. ويتاح التغيير على الوصول إلى النسخ عند الكتابة لأنه لا يؤثّر على المعالجات الأخرى التي تشارك البيانات.

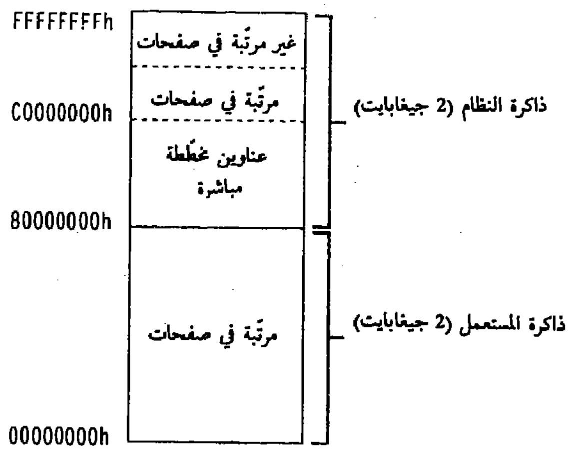
كذلك يستعمل الأمان عندما تنشىء شعبة قسم ليحتوي على ملفّ تخطيطي. للقيام بذلك، يجب أن تتمكّن الشعبة من الوصول إلى كائن الملفّ الأساسي. فمثلاً، يجب أن تتمكّن الشعبة التي تنشىء كائن قسم لتخطيط ملفّ من الوصول إلى قراءة الملفّ على الأقلّ وإلا تخفق العمليّة. بعد تحميل ملف في قسم، تستطيع الشعبة تغيير لائحة ACL على كائن القسم لكن ضمن الحدود المحدّدة من قبَل لائحة ACL على الملفّ المخطّط.

3-6 إستخدام الذاكرة الظاهريّة:

لقد ركّز هذا الفصل إلى الآن على مبادىء الذاكرة الظاهريّة العامّة ومزايا نمط المستعمل المتوفّرة في برنامج إدارة VM في البرنامج التنفيذي NT. تعالج الأقسام الفرعيّة التالية المسائل الداخليّة ـ لبنيات البيانات واللوغاريتمات غير المرئيّة لشيفرة نمط المستعمل، لكن التي تؤثّر على تشغيل وأداء الذاكرة الظاهريّة. يتمّ أولاً وصف تصميم فسحة العنوان الظاهري لمعالجة ويتبعه شرح عن آليات تعيين الصفحات والسياسات التي تتحكّم باستعمال المعالجة للذاكرة. يتبع ذلك وصف موجز لبنيتي البيانات الأساسيّتين في مكوّن الذاكرة الظاهريّة. وأخيراً تعالج مسائل المستوى الأعلى في المعالجة المتعدّدة والنقليّة لنظام الذاكرة الظاهريّة.

1-3-6 فسحة العنوان:

تحتوي كل معالجة NT محليّة على فسحة عنوان ظاهري كبيرة من 4 جيغابايت يحجز 2 جيغابايت منها لإستعمال النظام. ويمكن من النصف الأسفل من فسحة العنوان الظاهري الوصول إلى شعب نمط المستعمل ونمط النواة وهو فريد لكل معالجة. ومن النصف العلوي لفسحة العنوان الظاهري يتمُّ الوصول إلى شعَب نمط النواة وهي نفسها لكل معالجة. توضّح فسحة العنوان الظاهري لمعالجة في الشكل (6-8).



الشكل (6-8)
فسحة العنوان الظاهري

تستقرُّ شيفرة النواة والبيانات في الجزء الأسفل من ذاكرة النظام (من 80000000 إلى BFFFFFFFh على المعالج MIPS R4000)، ولا يتمُّ إخراجها من الذاكرة. وعلى المعالج MIPS R4000، تخطط مباشرة منطقة الذاكرة هذه من قبل العتاد. أي، يصفّر المعالج ثلاث بتات الأكثر أهميّة من أي عنوان ظاهري في هذا المجال ويستعمل البتات المتبقّية كعنوان فعلي (الذي يضع البيانات في ذاكرة فعليّة منخفضة). ولأن العناوين في هذا المجال تترجم من قبل العتاد وهي غير صالحة، فإن الوصول إلى البيانات من مجال الذاكرة هذا سريع جداً. وهو يستعمل لأجزاء من النواة التي تعتمد على أداء أقصى، مثل الشيفرة التي توزّع الشعَب للتنفيذ على معالج.

يتمُّ التحكُّم بالجزء الأعلى من ذاكرة النظام بواسطة برنامج إدارة VM وهو يستعمل لتخزين شيفرة وبيانات النظام الآخر. ويحجز جزءًا من هذه المنطقة للشيفرة والبيانات التي يمكن

ترتيبها في صفحات على قرص ويحجز الجزء الآخر لشيفرة النظام التي لا يمكن إزالة صفحاتها من الذاكرة (الشيفرة التي لا ترتّب في صفحات).

عند إنشاء معالجة جديدة، يمكن تحديد قيام برنامج إدارة VM بتحفيز فسحة عنوانه الظاهري عن طريق إستنساخ فسحة العنوان الظاهري لمعالجة أخرى أو بتخطيط ملفّ في فسحة العنوان الظاهري العائدة له. فمثلاً، يستعمل النظام الفرعي POSIX الطريقة السابقة عندما ينشىء أحد مستضافاته معالجة تابع. وتكون فسحة عنوان معالجة التابع نسخة عن معالجة الأم. (في الواقع يشارك الأم والتابع صفحات النسخ عند الكتابة، لذلك لا يتمّ أي نسخ فوري). وتستعمل هذه الطريقة الأخيرة عند إنشاء معالجة جديدة لتشغّل برنامج تنفيذي. فمثلاً، عندما يشغّل مستعمل البرنامج الخدماتي Chkdsk، ينشىء برنامج إدارة المعالجة NT معالجة ويحفّز برنامج إدارة VM فسحة عنوانها برسم Chkdsk الذي ينفّذ بعد ذلك.

يمكن أن تقدّم الأنظمة الفرعيّة للمحيط لمعالجات المستضاف مشاهد عن الذاكرة التي لا توافق فسحة العنوان الظاهري لمعالجة NT محليّة. تستعمل تطبيقات Win 32 فسحة عنوان مطابقة لفسحة العنوان المحلّي، لكن النظام الفرعي 16-bit OS/2 وماكنات DOS الظاهريّة (VDM) تقدّم مشاهدة معدّلة للذاكرة إلى مستضافاتها.

2-3-6 الترتيب في صفحات:

يكشف تصميم مكوّنات نظام التشغيل نفسه في غالب الأحيان عندما تسأل سؤالين مهمّين:

- ما هي الآليات التي تستعملها المكوّنات للقيام بعملها؟
- ما هي السياسات التي تتحكّم بالآليات؟

تتضمّن آليات الذاكرة الظاهريّة طريقة ترجمة برنامج إدارة VM للعناوين الظاهريّة إلى عناوين فعليّة وطريقة جلب الصفحات إلى الذاكرة الفعليّة. لكن من الناحية المقابلة، تحدّد سياسات الذاكرة الظاهريّة متى تجلب صفحة إلى الذاكرة ومكان وضعها.

يوفّر المعالج في غالب الأحيان آليات ترتيب الصفحات الأساسيّة التي يزيدها نظام الذاكرة الظاهريّة. ويعتبر ناقل الصفحات، أي شيفرة برنامج إدارة VM التي تنقل الصفحات إلى القرص ومنه، وسيط مهمّ بين آليات العتاد وسياسات البراجيّات:

- يجعل صفحة غير صالحة عند حصول خطأ صفحة (مثلاً، بتحميل صفحة في الذاكرة من قرص).

■ يوفّر حماية معتمدة على الصفحة للصفحات غير الصالحة ويحسّن الحمايات التي توفّرها العتاد للصفحات الصالحة.
■ يحدّث بنيات بيانات إدارة الذاكرة ويحافظ عليها.

إضافة لذلك، يفرض ناقل الصفحات سياسات الترتيب في الصفحات الموضوعيّة من قبَل برنامج إدارة VM. يصف القسم الفرعي التالي آليات الذاكرة الظاهريّة المزوّدة من قبَل MIPS R4000. ويلخّص القسم الفرعي الذي يليه سياسات الترتيب في صفحات لبرنامج إدارة VM.

1-2-3-6 آليات الترتيب في صفحات:

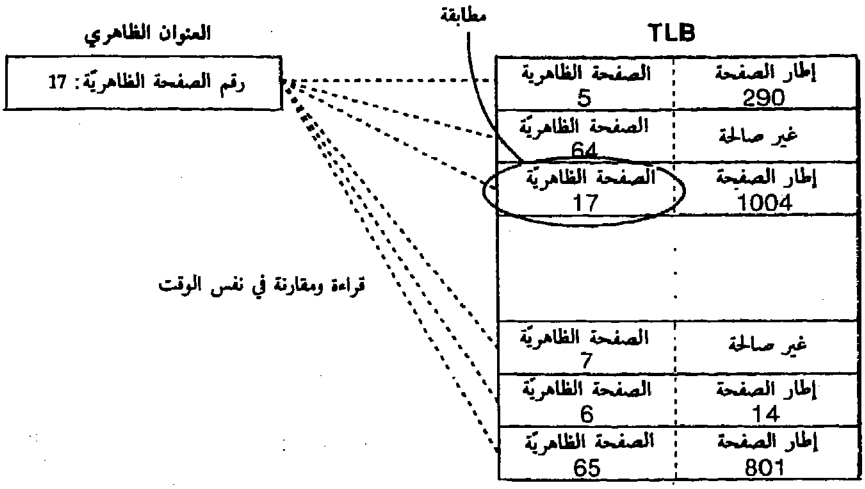
كل معالج يدعم ذاكرة ظاهريّة يقوم بذلك بطريقة مختلفة. لذلك، فإن الشيفرة التي تتداخل مباشرة مع الذاكرة الظاهريّة غير نقّالة ويجب تعديلها لكل منصّة عتاد جديدة. وفي أفضل الأحوال، كما في Windows NT، تكون هذه الشيفرة صغيرة ومعزولة.

إن المعلومات في هذا القسم خاصة بالمعالج MIPS R4000 وهي توفّر مثالاً حول كيفيّة إشتغال برنامج إدارة VM داخلياً مع معالج. وتنطبق معظم هذه المعلومات على معالجات Intel CISC لكن بغية التبسيط، لن يتمّ شرح معالجات Intel بشكل مطوّل هنا.

يحتوي المعالج MIPS R4000 على منظومتين؛ وحدة معالجة 32-bit RISC (تسمى CP1) ومنظومة على رقيقة مستقلّة (تسمى CP0) تتناول ترجمة العناوين والمناولة الإستثنائيّة. -تلتقط CP0 تلقائياً كل عنوان ينشئه برنامج وتترجمه إلى عنوان فعلي. فإذا كانت الصفحة التي تحتوي العنوان صالحة (موجودة في الذاكرة)، تحدّد CP0 موقعها وتستردّ المعلومات. وإذا كانت الصفحة غير صالحة (غير موجودة في الذاكرة)، تنشىء CP0 خطأ صفحة ويحفّز ناقل صفحات برنامج إدارة VM.

لضمان الوصول السريع إلى الذاكرة، يوفّر المعالج MIPS R4000 (ومعالجات Intel أيضاً) صفيفة من الذاكرة المتعلّقة تسمى المخزن المؤقّت الجانبي للترجمة (TLB). والذاكرة المتعلّقة، مثل TLB، هي متّجهة ذات خلايا يمكن قراءتها في نفس الوقت ومقارنتها مع قيمة هدف. وفي حالة TLB، تحتوي المتّجهة مخططات الصفحة الظاهريّة إلى الفعليّة لمعظم الصفحات المستعملة مؤخراً ونوع حماية الصفحة المطبّقة على كل صفحة. الشكل (6-9) وهو وصف مبسّط للمخزن المؤقّت TLB.

تحتوي العناوين الظاهريّة المستعملة بكثرة على إدخالات في TLB توفّر ترجمة عنوان ظاهري إلى فعلي سريعة جداً وبالتالي الوصول إلى الذاكرة. وإذا لم يكن العنوان الظاهري



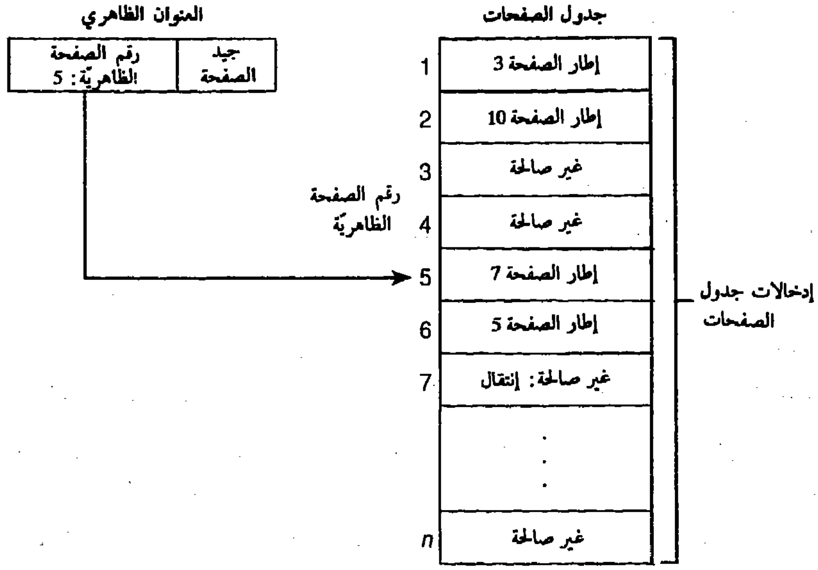
الشكل (6-9)

الوصول إلى المخزن المؤقّت الجانبي للترجمة

متواجداً في TLB، قد يكون في الذاكرة، لكن يجب على براجيات الذاكرة الظاهريّة، بدلاً من العتاد، إيجاده حيث يكون وقت الوصول أبطأ قليلاً. وإذا أُزيلت صفحة ظاهريّة من الذاكرة، يجعل نظام الذاكرة الظاهريّة إدخال TLB غير صالح. وإذا تمكّنت المعالجة من الوصول إليه مجدّداً، يحصل خطأ صفحة ويجلب برنامج إدارة VM الصفحة إلى الذاكرة ويعاود إنشاء إدخال في TLB.

تستعمل النواة وبرنامج إدارة VM جداول صفحات منشأة بواسطة البراجيات لإيجاد الصفحات غير الموجودة في TLB. تتواجد جداول الصفحات على معظم أنظمة الذاكرة الظاهريّة وهي تستخدم أحياناً من قبل العتاد وفي بعض الأحيان من قبل البراجيّات. وفي الناحية المبدئيّة، يشبه جدول الصفحات بنية البيانات المبيّنة في الشكل (6-10).

يحتوي إدخال جدول صفحات (PTE) كل المعلومات الضروريّة لكي يحدّد نظام الذاكرة الظاهريّة موقع صفحة عندما تستعمل شعبة عنواناً. وفي نظام الذاكرة الظاهريّة البسيط، فإن الإدخال غير الصحيح في جدول صفحات يعيّن عدم وجود الصفحة في الذاكرة الفعليّة ويجب تحميلها من القرص. يحصل إستثناء لخطأ الصفحة وتحمّل براجيات ترتيب الصفحات الصفحة المطلوبة إلى الذاكرة وتحدّث جدول الصفحات. يعاود المعالج إصدار التعليمات التي تنشىء خطأ الصفحة. لكن في هذا الوقت يكون إدخال جدولة الصفحات صالحاً وتسترد البيانات بنجاح من الذاكرة.



الشكل (6-10)
جدول الصفحات المفاهيمي

يتّصف المعالج MIPS R4000 بعناوين من 32 بت أو عناوين ظاهريّة ممكنة من 2^{32} لكل معالجة وهو ينظّم هذه العناوين الظاهريّة إلى صفحات بطول 2^{12} بايت (4 كيلوبايت) التي تؤدّي إلى صفحات 2^{20} أو 1,048,576 لكل فسحة عنوان. وإذا كانت إدخالات جدول الصفحات بعرض 4 بايت، فإنها تستهلك 1024 إطار صفحة من التخزين ($2^{2} \times 2^{20}$ مقسومة على 2^{12}) لتخطيط كل الذاكرة الظاهريّة. وهذا لفسحة عنوان واحد فقط. تحتوي كل معالجة على فسحة عنوان مستقلّ. ولتجنُّب إستهلاك كل الذاكرة فقط لجداول الصفحات، يدخل برنامج إدارة VM صفحات الجداول ويخرجها من الذاكرة عند طلبها.

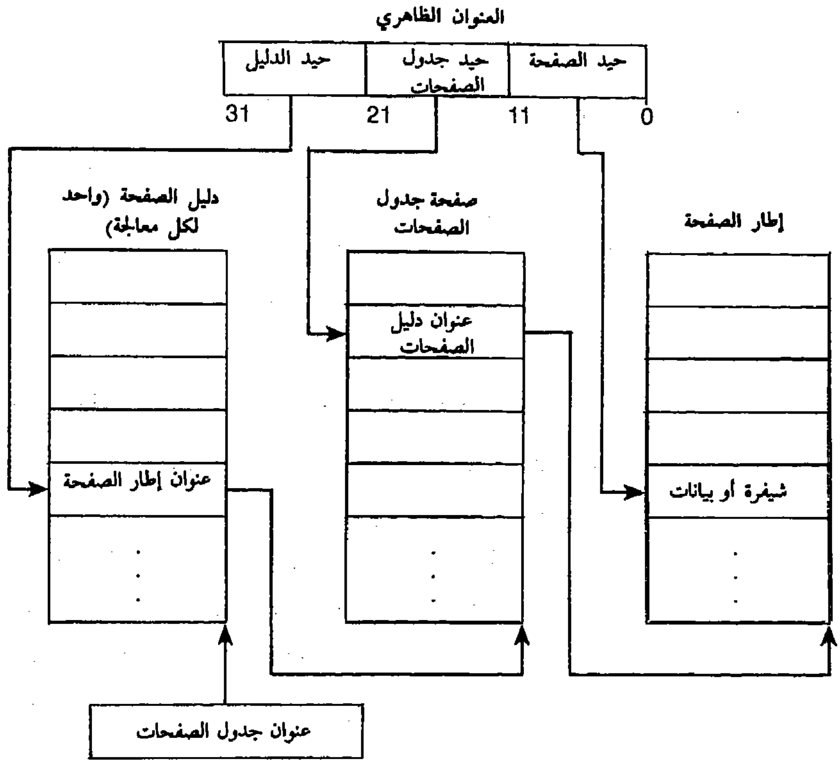
يتيح المعالج MHPS R4000 لنظام التشغيل إنشاء جداول صفحات في النسق المناسب. وبالمقابل، ينشىء المعالج Intel 386 نسق جدول صفحات في العتاد. ولتوفير النقليّة القصوى من معالجات MIPS إلى Intel، يعتمد برنامج إدارة VM بنية جدول صفحات من مستويين يحاكي نسق Intel. يشير جدول المستوى الأول الذي يسمى دليل الصفحة، إلى الصفحات في جدول صفحات المستوى الثاني. ويشير جدول صفحات المستوى الثاني إلى أُطر الصفحة الفعليّة كما يبينّ في الشكل (6-11) على الصفحة التالية.

عند تحديد موقع إدخال جدول صفحات، يترجم برنامج إدارة VM (والنواة NT) عنوان ظاهري بشكل MIPS إلى عنوان بشكل Intel بإستعمال أجزاء مختلفة منه كحيد في بنية جدول

الصفحات. إضافة لذلك، يحتوي دائماً إدخال واحد في TLB العنوان المرجعي الظاهري لدليل الصفحة للمعالجة المنفّذة حالياً. (هذا هو سبب عدم مشاهدة معالجة مستعمل واحد لفسحة عنوان الآخر. وهي تحتوي على أدلّة صفحات مختلفة تشير إلى جداول صفحات مختلفة).

يمكن للإدخالات في دليل صفحات المعالجة وجداول الصفحات أن تكون إما صالحة أو غير صالحة. فإذا كان الإدخال في دليل الصفحات غير صالح، يحصل خطأ صفحة آخر لتحديد موقع الشيفرة أو صفحة البيانات.

إن إدخال جدول الصفحات في جداول الصفحات المعرّفة من قبَل NT هي تحسين على جدول الصفحات المفاهيمي المبيّن سابقاً. يحتوي كل إدخال جدول صفحات (وكل إدخال دليل صفحات) على علم إنتقال، وإذا علّم إدخال جدول الصفحات على أنه غير صالح وضبط علم الإنتقال، تسجّل الصفحة لمعاودة الإستعمال، لكن محتوياتها تبقى صالحة. إن جعل صفحة إنتقاليّة صالحة هو عمليّة سريعة جداً لأي برنامج إدارة VM لا يحتاج لقراءة الصفحة في الذاكرة



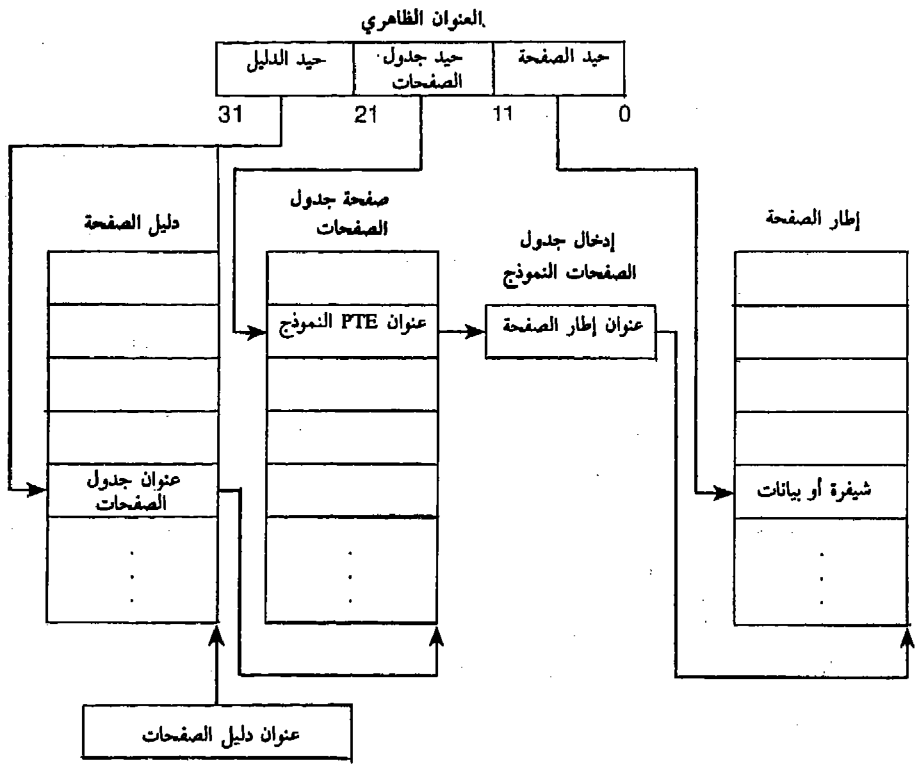
الشكل (6-11)

بنية جدول الصفحات على معالجات Intel 386 و MIPS R40006

من قرص. كذلك تحتوي إدخالات جدول الصفحات على أعلام تسجّل حمايات مستوى الصفحة التي يطبّقها برنامج إدارة VM على كل صفحة.

عند مشاركة إطار صفحة بين معالجتين، يدرج برنامج إدارة VM مستوى لا إتجاهي في جداول الصفحات العائدة له، كما يوضّح ذلك الشكل (6-12). تسمّى بنية البيانات التي يدرجها إدخال جدول الصفحات النموذج (PTE نموذج).

يتيح PTE النموذج، وهو بنية من 32 بت تشبه إدخال جدول صفحات عادي، لبرنامج إدارة VM إدارة الصفحات المشاركة دون الحاجة لتحديث جداول صفحات كل معالجة تشارك الصفحة. فمثلاً، يمكن إزالة صفحة شيفرة مشاركة أو صفحة بيانات إلى قرص عند نقطةٍ ما. وعندما يستردّ برنامج إدارة VM الصفحة من القرص (يدخل الصفحة)، عليه أن يحدث المؤشّر المخزّن في PTE النموذج للإشارة الموقع الفعلي الجديد للصفحة. ويبقى جدول صفحات المعالجة



الشكل (6-12)
ترجمة العنوان للذاكرة المشاركة

المشاركة هو نفسه. ويتمُّ توزيع إدخالات PTE النموذج من فسحة النظام المرتّب في صفحات، لكي يمكن ترتيبها في صفحات مثل إدخالات جدول الصفحات، عند الضرورة.

6-3-2-2 سياسات الترتيب في صفحات ومجموعات العمل:

تعرّف عادة أنظمة الذاكرة الظاهريّة ثلاث سياسات تحدّد كيفيّة تنفيذ عمليّة الترتيب في صفحات: سياسة الجلب وسياسة الوضع وسياسة الإستبدال.

تحدّد سياسة الجلب عندما يجلب ناقل الصفحات صفحة من قرص إلى ذاكرة. يحاول أحد أنواع سياسة الجلب تحميل الصفحات المطلوبة من قبل معالجة قبل أن تطلبها. تحمّل سياسة جلب أخرى، تسمى سياسات الترتيب في صفحات للطلبات، صفحة في الذاكرة الفعليّة فقط عند حصول خطأ صفحة. وفي نظام الترتيب في صفحات للطلبات، تتعرّض المعالجة لعدّة أخطاء صفحة عندما تبدأ شعَبها التنفيذ لأول مرة، عندما تشير إلى المجموعة الأوليّة للصفحات المطلوب متابعتها. وبعد تحميل مجموعة الصفحات هذه في الذاكرة، ينخفض نشاط المعالج في ترتيب الصفحات.

يستعمل برنامج إدارة VM في NT لوغاريتم ترتيب الصفحات للطلبات مع «تجميع عنقودي» لتحميل الصفحات في الذاكرة. وعندما تحصل شعبة على خطأ صفحة، يحمّل برنامج إدارة VM في الذاكرة الصفحة الخطأ إضافة إلى عدد من الصفحات التي تحيط بها. تحاول هذه الطريقة تخفيض عدد أخطاء الصفحة التي تتعرّض لها الشعبة. ولأن البرامج، وخاصة الكبيرة منها، تحاول التنفيذ في مناطق صغيرة من فسحة عنوانها في أي وقت، فإن تحميل مجموعات عنقوديّة من الصفات الظاهريّة يخفّض عدد أخطاء الصفحة.

عندما تستلم شعبة خطأ صفحة، يجب أن يحدّد نظام إدارة الذاكرة أيضاً مكان وضع الصفحة الظاهريّة في الذاكرة الفعليّة. تسمى مجموعة القواعد التي تتّبعها سياسة الوضع. إن سياسات الوضع، رغم كونها معقّدة في غالب الأحيان لتصاميم الذاكرة المقطعيّة، هي عادة بسيطة للتصاميم الخطيّة، التي تتطلّب فقط إنشاء إطار صفحة خالية. وفي NT، وإذا لم تكن الذاكرة ممتلئة، ينتقي برنامج إدارة VM إطار الصفحة الأولى على لائحة أُطر الصفحة الفارغة. وإذا كانت اللائحة فارغة، فإنها تعترض سلسلة من لوائح أُطر الصفحات الأخرى التي تحافظ عليها. ويعتمد ترتيب الإعتراض على نوع خطأ الصفحة الحاصل. (توجد معلومات إضافيّة حول لوائح أُطر الصفحات في القسم 6-3-3).

إذا كانت الذاكرة الفعليّة ممتلئة عند حصول خطأ صفحة، تستعمل سياسة إستبدال

لتحديد الصفحة الظاهريّة الواجب إزالتها من الذاكرة لإفساح المجال لصفحة جديدة. تشمل سياسات الإستبدال العامة أقلّ صفحة إستُعملت مؤخراً (LRU) والصفحة الأولى الداخل والأولى الخارجة (FIFO). يتطلّب لوغاريتم LRU من نظام الذاكرة الظاهريّـة تعقّب متى تمّ إستعمال صفحة في الذاكرة. وعند طلب إطار صفحة جديدة، ترسل الصفحة التي لم يتمّ إستعمالها لفترة زمنيّة طويلة إلى القرص ثم إخلاء إطارها وفقاً لخطأ الصفحة. أما لوغاريتم FIFO فهو أسهل ويزيل الصفحة التي كانت في الذاكرة الفعليّة لأكبر فترة زمنيّة، دون إعتبار لكميّة إستعمالها.

يمكن تصنيف سياسات الإستبدال على أنها إما شاملة أو محليّة. تحدّد سياسة إستبدال محلّي عدداً ثابتاً (أو كما في NT، عدداً قابلاً للتعديل دينامياً) من أُطر الصفحات لكل معالجة. وعندما تستعمل معالجة كل حصتها، تخلي براجميات الذاكرة الظاهريّة (أي، تزيل من الذاكرة الفعليّة) إحدى صفحاتها لكل خطأ صفحة جديدة يعترضها. تتيح سياسة الإستبدال الشاملة إجابة خطأ صفحة من قبل أي إطار صفحة، دون إعتبار لكون هذا الإطار عائد لمعالجة أخرى. فمثلاً، تحدّد سياسة إستبدال شاملة تستعمل لوغاريتم FIFO الصفحة التي كانت في الذاكرة لأطول مدّة وتخليها إجابة لخطأ صفحة. تحصر سياسة الإستبدال المحليّة بحثها عن الصفحة الأقدم على مجموعة الصفحات العائدة للمعالجة التي جلبت خطأ الصفحة.

تتّصف سياسات الإستبدال الشاملة بعدّة علل. أولاً، فهي تجعل المعالجات عرضة لتصرُّف المعالجات الأخرى. فمثلاً، إذا كانت معالجة واحدة أو أكثر في النظام تستعمل كميات كبيرة من الذاكرة، يحتمل أن يجلب تطبيق منفذ صفحات كثيرة. ويزداد وقت التنفيذ. ثانياً، يستطيع تطبيق سيّىء التصرُّف تقويض نظام التشغيل بأكمله عن طريق تحفيز نشاط زائد في تحديد الصفحات في كل المعالجات. وفي Windows NT، من المهمّ جداً عدم منافسة الأنظمة الفرعيّة للمحيط مع المعالجات الأخرى على حصتها من الذاكرة. ويجب أن يحافظوا على عدد معيّن من الصفحات في الذاكرة للتنفيذ بشكل كافي ودعم تطبيقات المستضاف بشكل مناسب. لهذه الأسباب، يستعمل برنامج إدارة VM سياسة إستبدال FIFO محليّة. يتطلّب هذا الإجراء متابعة برنامج إدارة VM تعقّب الصفحات الموجودة حالياً في الذاكرة لكل معالجة. تسمى مجموعة الصفحات الجديدة هذه مجموعة عمل المعالجة.

عند إنشائها، يعيّن لكل معالجة حجم مجموعة عمل أدنى، الذي هو عدد الصفحات المضمون توفّرها للمعالجة في الذاكرة خلال إشتغالها. وإذا لم تكن الذاكرة ممتلئة، يتيح برنامج إدارة VM للمعالجة الحصول على قدر ما يحدّده الحجم الأقصى لمجموعة العمل من الصفحات.

فإذا إحتاجت المعالجة لصفحات إضافيّة، يزيل برنامج إدارة VM إحدى صفحات المعالجة لكل خطأ صفحة جديدة تنشئه المعالجة.

لتحديد الصفحة الواجب إزالتها من مجموعة العمل لمعالجة، يستخدم برنامج إدارة VM لوغاريتم FIFO بسيطاً، حيث يزيل الصفحات التي تواجدت في الذاكرة لفترة طويلة (ولأن صفحات مجموعة العمل المستبدلة تبقى فعلياً في الذاكرة الفعليّة لفترة من الوقت بعد الإستبدال، فإنه يمكن إرجاعها إلى مجموعة العمل بسرعة دون الحاجة لعملية قراءة قرص. راجع القسم 6-3-3).

عندما تبدأ الذاكرة الفعليّة بالإمتلاء، يستعمل برنامج إدارة VM طريقة تسمى التهذيب التلقائي لمجموعة العمل لزيادة كميّة الذاكرة الحالية المتوفّرة في النظام. تفحص هذه الطريقة كل معالجة في الذاكرة حيث تقارن الحجم الحالي لمجموعة عملها مع القيمة الدنيا لمجموعة العمل. وعندما تجد معالجة تستعمل أكثر من حدّها الأدنى، فإنها تزيل الصفحات من مجموعات العمل العائدة لها حيث توفّر الصفحات لإستعمالات أخرى. وإذا بقيت كميّة الذاكرة الحالية منخفضة، يتابع برنامج إدارة VM إزالة الصفحات من مجموعات عمل المعالجات إلى أن تحصل كل معالجة على الحدّ الأدنى لمجموعة العمل العائدة لها.

عندما تهبط معالجة إلى الحدّ الأدنى لمجموعة العمل، يتعقّب برنامج إدارة VM عدد أخطاء الصفحة التي تجلبها المعالجة. فإذا أدّت المعالجة إلى أخطاء صفحة ولم تكن الذاكرة ممتلئة، يزيد برنامج إدارة VM حجم مجموعة عمل المعالجة. لكن إذا لم تجلب المعالجة أخطاء صفحة لفترة من الوقت، فالسبب يعود إما إلى ملاءمة الشيفرة التي تنفّذها شعَب المعالجة بشكل مريح ضمن مجموعة العمل الدنيا للمعالجة أو إلى عدم تنفيذ أي من شعَب المعالجة. فمثلاً، تنتظر معالجة التسجيل في Windows NT قيام مستعمل بالتسجيل. وبعد تسجيل المستعمل، تنتظر المعالجة إنهاء المستعمل. بالنسبة لمعالجة التسجيل والمعالجة الأخرى التي تبقى متوقّفة معظم الوقت، يتابع برنامج إدارة VM تخفيض مجموعة عمل المعالجة إلى أن تجلب المعالجة خطأ صفحة. ويشير خطأ الصفحة إما إلى تحفيز شعَب المعالجة أو إلى بلوغ المعالجة الحدّ الأدنى من الذاكرة التي تحتاجها الشعَب للتنفيذ.

تستطيع معالجة تغيير الحدّ الأدنى والأقصى لمجموعة العمل باستدعاء خدمة كائن معالجة، لكن قاعدة بيانات السياسة المحليّة لنظام الأمان تضبط حداً أدنى وأقصى لكل معالجة في نمط المستعمل. ورغم توفّر هذه القدرة، من غير الضروري للمعالجات الإفراديّة تعديل قيم مجموعة العمل العائدة لها. فقد صمّم برنامج إدارة الذاكرة، عبر إستعماله لسياسة الإستبدال المحليّة

والتهذيب التلقائي لمجموعة العمل، ليتعقّب الحمل على الذاكرة ولكي يعدّل إستعمال الذاكرة وفقاً لذلك. وهو يحاول توفير أفضل أداء ممكن لكل معالجة دون الحاجة لضبط النظام من قبَل المستعملين الإفراديين أو من قبَل مدير.

3-3-6 قاعدة بيانات إطار الصفحة:

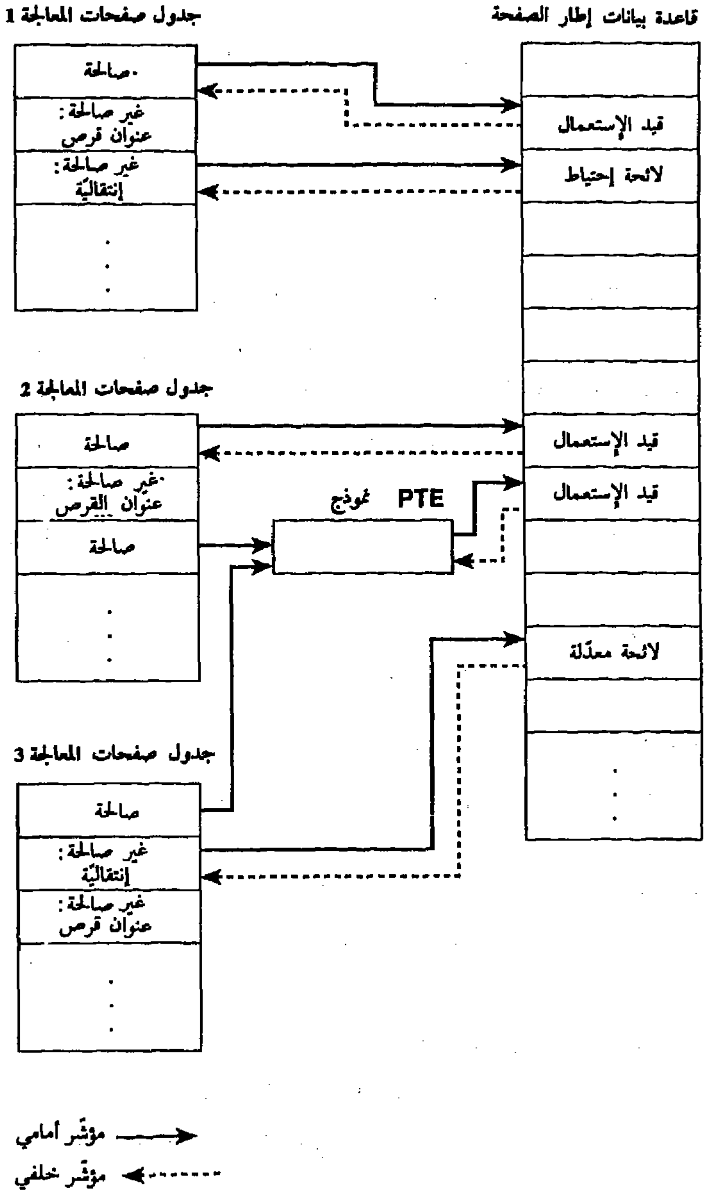
تتعقّب جداول صفحة معالجة حيث تخزّن صفحة ظاهريّة في الذاكرة الفعليّة. كذلك يحتاج برنامج إدارة VM إلى بنية بيانات لتعقّب حالة الذاكرة الفعليّة. فمثلاً، فإنه يحتاج لتسجيل فيما إذا إطار صفحة طليق وإذا لم تكن كذلك، فمن يستعملها. تجيب قاعدة بيانات إطار الصفحة هذا الطلب. وهي صفيفة إدخالات مرقّمة من 0 إلى عدد أُطُر صفحات الذاكرة في النظام (ناقص 1). يحتوي كل إدخال معلومات تتعلّق بإطار الصفحة الموافق. تصوّر قاعدة بيانات إطار الصفحة وعلاقتها مع جداول الصفحات في الشكل (6-13). وكما يظهر هذا الشكل، تشير إدخالات جدول الصفحات الصالحة إلى الإدخالات في قاعدة بيانات إطار الصفحة وتشير إدخالات قاعدة بيانات إطار الصفحة إلى جدول الصفحات الذي يستعملها. يستعمل برنامج إدارة VM المؤشّر الأمامي عندما تتمكّن معالجة من الوصول إلى عنوان ظاهري صالح. وهو يتبع المؤشّر لتحديد موقع الصفحة الفعليّة الموافقة للعنوان الظاهري.

كذلك تشير بعض إدخالات جدول الصفحات غير الصالحة إلى الإدخالات في قاعدة بيانات الصفحة. وتشير إدخالات جدول الصفحات «الإنتقاليّة» إلى أُطر الصفحات الصالحة للإستعمال والتي لم يتم معاودة إستعمالها بعد، وبالتالي ما زالت متماسكة في الذاكرة. وإذا تمكّنت المعالجة من الوصول إلى إحدى هذه الصفحات قبل معاودة إستعادة من قبل معالجة أخرى، يستطيع برنامج إدارة VM إستعادة المحتويات بسرعة.

تحتوي إدخالات جدول الصفحات غير الصالحة الأخرى عنوان القرص حيث تخزّن الصفحة. وعندما تتمكّن معالجة من الوصول إلى إحدى هذه الصفحات، يحصل خطأ صفحة ويقرأ برنامج إدارة VM الصفحة من القرص.

يمكن لأطر الصفحات أن تتواجد في حالة واحدة من ستّ حالات في أي وقت:

- صالح: إطار الصفحة قيد الإستعمال من قبل معالجة ويشير إليه إدخال جدول صفحات صالح.
- مصفّر: إطار الصفحة خالياً وتمّ تحفيزه بالأصفار.
- خالي: إطار الصفحة خالٍ لكنّه غير محفّز.



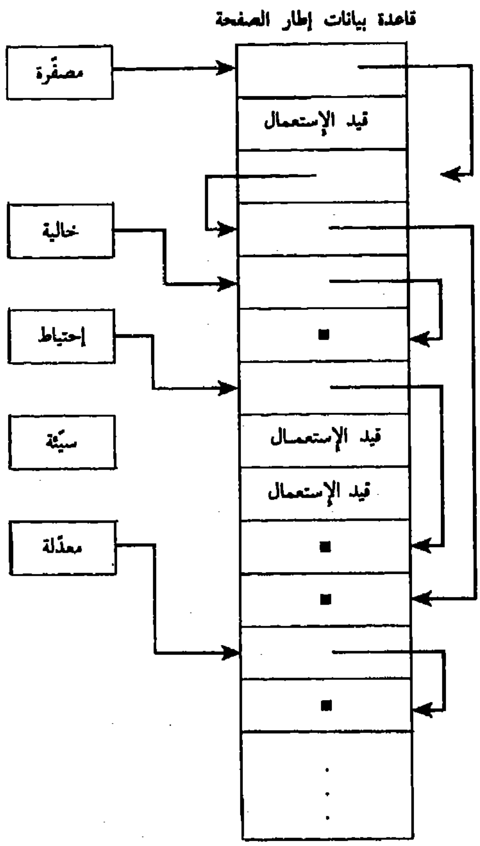
الشكل (6-13)

جداول الصفحات وقاعدة بيانات إطار الصفحة

■ إحتياط: إستعملت معالجة إطار صفحة، لكن إطار الصفحة أُزيل من مجموعة عمل المعالجة. وإدخال جدول الصفحات لها غير صالح لكنّه معلّم بعلم إنتقالي.

■ معدّل: هذه الحالة هي نفس حالة الإحتياط بإستثناء أن المعالجة التي إستعملت الصفحة كتبت إليها أيضاً، ولم تكتب المحتويات إلى القرص. وإدخال جدول الصفحات غير صالح لها لكنّه معلّم بعلم إنتقالي.
■ سيّىء: أنشأ إطار الصفحة إزدواجيّة أو أخطاء عتاد أخرى ولا يمكن إستعماله.

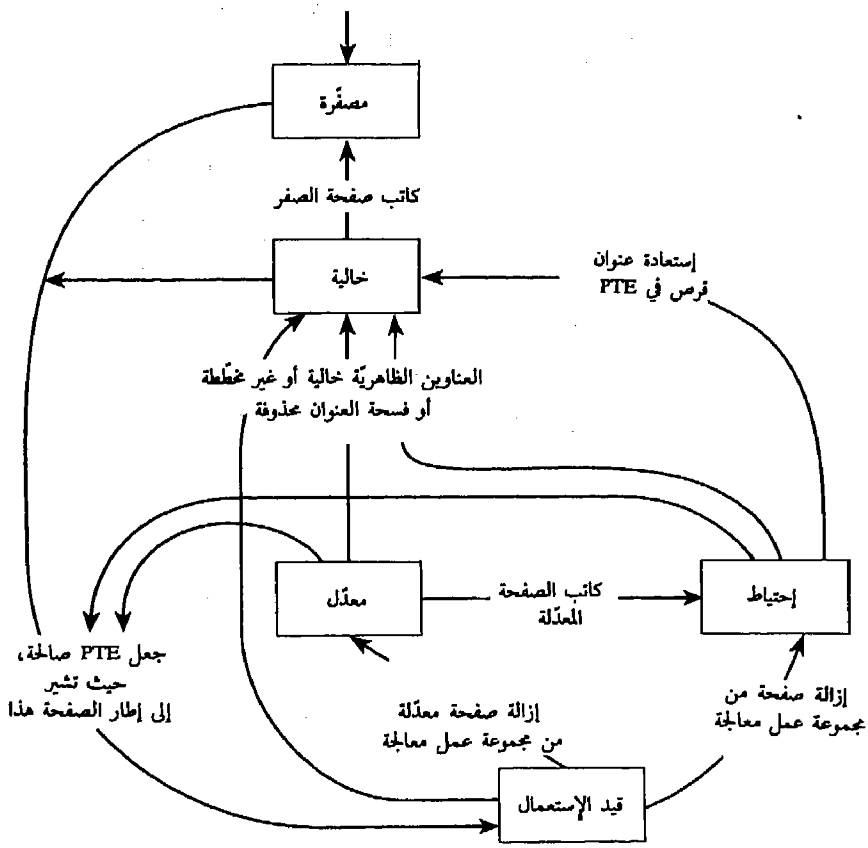
ومن بين أُطر الصفحات غير المستعملة، تربط قاعدة بيانات إطار الصفحة سويّة كل تلك الموجودة في نفس الحالة، حيث تنشىء خمس لوائح مستقلّة: اللائحة المصفّرة واللائحة الخالية واللائحة الإحتياط واللائحة المعدّلة ولائحة الصفحة السيّئة. تظهر العلامة بين قاعدة بيانات إطار الصفحة ولوائح الصفحات في الشكل (6-14).



الصفحة (6-14)
لوائح الصفحات في قاعدة بيانات إطار الصفحة

كما يظهر في الشكل، تحسب هذه اللوائح كل إطار صفحة في الحاسوب التي ليست قيد الإستعمال. ويُشار إلى تلك قيد الإستعمال من قبل معالجة بواسطة جدول صفحات المعالجة. عندما تنتهي معالجة بإطار صفحة أو عندما يرتّب برنامج إدارة VM محتوياته إلى قرص، يصبح إطار الصفحة خالياً ويضعه برنامج إدارة VM مجدّداً في إحدى لوائح إطار الصفحة العائدة له.

عندما يحتاج برنامج إدارة VM لإطار صفحة محفّزة (واحد يحتوي أصفاراً) لخدمة خطأ صفحة، فإنه يحاول الحصول على أول واحد في لائحة الصفحة المصفّرة. وإذا كانت اللائحة فارغة، فإنه يجلب واحداً من اللائحة الخالية ويصفّره. وعندما لا يحتاج برنامج إدارة VM لصفحة مصفّرة، فإنه يستعمل واحداً من اللائحة الخالية وإذا كانت اللائحة فارغة، فإنه يستعمل واحد من اللائحة المصفّرة. وفي أي من هاتين الحالتين، يستعمل برنامج إدارة VM



الشكل (15-6)
مخطّط حالة أطر الصفحة

اللائحة الإحتياط إذا كانت اللائحتان الأخرتان فارغة. وعندما ينخفض عدد الصفحات في اللائحات المصفّرة والخالية والإحتياطيّة دون القيمة المشرفيّة الدنيا، تحفّز شعبة تسمى كاتب الصفحة المعدّلة وتكتب محتويات الصفحات المعدّلة إلى قرص، ثم تنقلها إلى لائحة إحتياط لمعاودة إستعمالها.

وإذا أصبحت لائحة الصفحة المعدّلة قصيرة جداً، يبدأ برنامج إدارة VM تهذيب كل مجموعة عمل لمعالجة على لائحة معدّلة أو إحتياط لمعاودة إستعمالها عند الطلب. يظهر مخطّط إطالة الإنتقاليّة لإطار الصفحة في الشكل (6-15) على الصفحة التالية.

وقبل أن يتمكّن برنامج إدارة VM من إستعمال إطار صفحة من اللائحة الإحتياط أو المعدّلة، عليه أولاً من تعقّب إدخال جدول الصفحات غير الصالحة وتحديثها (أو PTE النموذج) الذي يشير إلى إطار الصفحة. وبالعودة إلى الشكل (6-13)، يمكن مشاهدة أن الإدخالات في قاعدة بيانات إطار الصفحة تحتوي مؤشّرات خلفيّة إلى جدول صفحات المستعمل السابق (أو إلى PTE نموذج لصفحات مشاركة)، التي تتيح حصول هذا التحديث.

6-3-4 واصفات العنوان الظاهري:

وصفَ قسم سابق من هذا الفصل سياسات الترتيب في الصفحات التي يستعملها برنامج إدارة VM ليحدّد متى يجلب صفحة إلى الذاكرة ومكان وضعها والصفحات الواجب إزالتها عندما تمتلىء الذاكرة.

يستعمل برنامج إدارة VM لوغاريتم ترتيب صفحات الطلب لمعرفة متى تحمّل الصفحات في الذاكرة. وهو ينتظر إلى أن تستعمل شعبة ما عنواناً وتجلب خطأ صفحة قبل أن تحصل على صفحة من القرص. إن ترتيب صفحات الطلب كهذه هي شكل من أشكال التقييم الكسول. تتجنّب لوغاريتمات التقييم الكسول تنفيذ عملية مكلفة مثل الترتيب في صفحات، إلى أن يطلب ذلك.

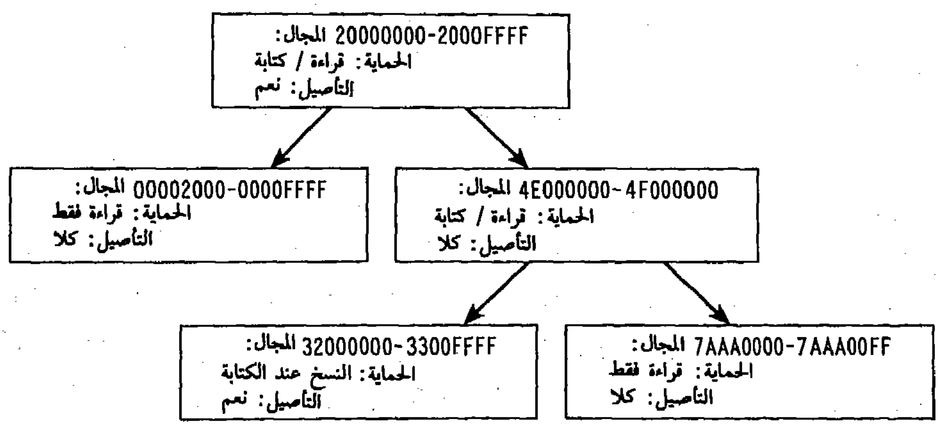
يستعمل برنامج إدارة VM التقييم الكسول في منطقة أخرى، أي إنشاء جداول صفحات. فمثلاً، عندما تحدّد شعبة منطقة كبيرة من الذاكرة الظاهريّة، يستطيع برنامج إدارة VM إنشاء جداول الصفحات المطلوبة فوراً للوصول إلى المجال الكامل للذاكرة المحدّدة. لكن، إذا لم يستعمل التطبيق كل الذاكرة المحدّدة، فإن إنشاء جداول الصفحات جهد دون فائدة. لذلك، ينتظر برنامج إدارة VM للقيام بذلك إلى أن تجلب شعبة لخطأ صفحة. يؤدي إستعمال التقييم الكسول بهذه الطريقة إلى كسب أداء كبير للتطبيقات التي تحجز كميّة ذاكرة كبيرة ولكن تستعمل القليل منها.

إن عمليّة تحديد الذاكرة وحتى الكتل الكبيرة منها، هي سريعة للغاية بواسطة لوغاريتم التقييم الكسول. لكن كسب الأداء ليس دون تناوب. فعندما تحدّد شعبة الذاكرة، يجب أن يستجيب برنامج إدارة VM بمجال من العناوين الظاهريّة لتستعملها الشعبة. لكن ولأن برنامج إدارة VM لا يحمّل جدول صفحات المعالجة إلى أن تتمكّن الشعبة فعلياً من الوصول إلى الذاكرة، فإنه لا يستطيع معاينة جدول الصفحات لتحديد العناوين الظاهريّة الحالية. لذلك، يجب أن يحافظ برنامج إدارة VM على مجموعة أخرى من بنيات البيانات لمتابعة تعقُّب العناوين الظاهريّة التي تمّ تمديدها في فسحة عنوان المعالجة وتلك لم يتم تحديدها بعد. تقوم واصفات العنوان الظاهري بهذه المهمة.

لكل معالجة، يحافظ برنامج إدارة VM على مجموعة من واصفات العنوان الظاهري التي تصف حالة فسحة العنوان الظاهري المعالجة. راجع الشكل (6-16).

عندما تحدّد معالجة الذاكرة (أو تخطّط مشهد ذاكرة مشاركة)، ينشىء برنامج إدارة VM واصف عنوان ظاهري لتخزين أية معلومات مزوّدة في طلب التحديد مثل مجال العناوين المحدّدة، لجهة كون المجال ذاكرة مشاركة أو خاصة ولجهة إمكانيّة معالجة تابع تأهّل محتويات المجال وحماية الصفحة المطبقة على الصفحات في المجال. ثم يدرج واصف العنوان الظاهري في شجرة الإنحدار الخاصة بمعالجة (شجرة ثنائيّة ذاتيّة الموازنة) لتسريع معالجة تحديده.

عند تمكّن معالجة لأول مرة الوصول إلى عنوان، يجب على برنامج إدارة VM إنشاء إدخال جدول صفحات للصفحة التي تحتوي العنوان. وللقيام بذلك، يجد برنامج إدارة VM واصف



الشكل (6-16)
واصفات العنوان الظاهري

العنوان الظاهري الذي يحتوي مجال عنوانه العنوان الذي تمَّ الوصول إليه ويستعمل المعلومات التي يجدها لتعبئة إدخال جدول الصفحات. وإذا كان العنوان خارج المجال المغطّى من قبَل واصف العنوان الظاهري، يعرف برنامج إدارة VM أن الشعبة لم تحدّد عنوان الذاكرة هذا قبل إستعماله، حيث حصل مخالفة في الوصول.

5-3-6 إعتبارات المعالجة المتعدّدة:

تتعرّض أية شيفرة يمكن أن تشتغل على أكثر من معالج واحد في نفس الوقت لبعض تقييدات التشفير. يجب أن تكون الشيفرة مشتركة ويجب أن تتيح لشعبة واحدة فقط الوصول إلى بنيات البيانات المشاركة في كل مرة، ويجب أن لا تسمح لشعبتين الحصول على الموارد بطريقة تمنع تنفيذ بعضها البعض (وهي حالة تُعرف بإسم الطريق المسدود). إضافة لذلك، تظهر إعتبارات الأداء في الأنظمة المتعدّدة المعالجات التي لا تحصل في أنظمة آحاديّة المعالج.

إن برنامج إدارة VM مشترك، لذلك منع تشويه البيانات والطريق المسدود وتحقيق إداء جيد هي أكثر الإعتبارات أهميّة لتنفيذ المعالجات المتعدّدة العائدة له.

يستعمل برنامج إدارة VM قفلاً دوّاراً لحماية أهم بنيات البيانات لديه ــ أية قاعدة بيانات إطار الصفحة. وعند حصول خطأ صفحة، يتحكّم ناقل الصفحات بالشعَب الخطأ كل خطأ الصفحة وتحديث قاعدة البيانات. وقبل الوصول إلى قاعدة البيانات، يجب أن تحتوي الشعبة على القفل الدوّار لقاعدة البيانات. وخلال إحتواء الشعبة على القفل الدوّار، لا تستطيع أية شعبة أخرى قراءة قاعدة بيانات إطار الصفحة أو الكتابة إليها. لذلك، عند حصول خطأ صفحة في نفس الوقت على Windows NT، يمكن تعليق شعبة واحدة مؤقّتاً إلى أن تصبح قاعدة بيانات إطار الصفحة خالية.

إن إعاقة وصول الشعبة إلى قاعدة بيانات إطار الصفحة هي مثال عن المشاركة بين السرعة وحاجات الأنظمة المتعدّدة المعالجة. وقد يتيح برنامج إدارة VM لشعب مستقلّة بالوصول إلى أجزاء مختلفة من قاعدة البيانات في نفس الوقت عن طريق تقسيم قاعدة البيانات إلى عدّة بنيات بيانات ووقاية كل جزء بقفل مستقلّ. لكن الحصول على أقفال إفلات هي عمليّة مكلفة، وتستغرق أخطاء الصفحة فترة أطول إذا إحتاجت الشعبة للحصول على ثلاث أقفال عِوَضاً عن قفل واحد للوصول إلى قاعدة البيانات. وبموازنة هذه المشاركة، فضّل Lou Perazzoli، مصمِّم برنامج إدارة VM، أخطاء الصفحة السريعة على التوازي المتزايد في ناقل الصفحات. وقد إختار إستعمال قفل واحد لحماية قاعدة البيانات، بإفتراض أنه مع العلو الأدنى، تدخل شعبة قاعدة البيانات وتخرج منها بسرعة أكبر وبالتالي تخلّيها للشعَب الأخرى.

إن إستعمال قفل قاعدة بيانات واحد يعني أنه يمكن لقاعدة بيانات إطار الصفحة أن تصبح عنق زجاجة أداء عندما يكون نشاط الترتيب في صفحات مرتفعاً. لتجنُّب هذه المشكلة، يحاول برنامج إدارة VM تخفيض أخطاء الصفحة إلى الحدّ الأدنى في Windows NT. وهو يقوم بما يلي لإبقاء عدد أخطاء الصفحة منخفضاً:

- يوفّر لكل معالجة ما يكفي من الصفحات في مجموعة العمل العائدة لها لمنع الخطأ الزائد.
- يهذّب تلقائياً مجموعات عمل المعالجات لتوفير الصفحات غير المستعملة أو الزائدة للمعالجات الأخرى.

6-3-6 إعتبارات النّقليّة:

يعتمد برنامج إدارة VM على مزايا عتاد معيّنة. إن ما يلي هو متطلّبات المعالج لبرنامج إدارة VM:

- عناوين 32 بت. (تدعم عناوين 64 بت لكنها تحتاج لإعادة تصميم في برنامج إدارة VM).
- دعم الذاكرة الظاهريّة والترتيب في صفحات. يجب أن يوفّر المعالج القدرة على تخطيط العناوين الظاهريّة إلى عناوين فعليّة ويجب أن تدعم آليات الترتيب في صفحات.
- الشفافيّة، مخابىء العتاد المتجانس للأنظمة المتعدّدة المعالجات. فعندما تحدّث شعبة على معالج واحد البيانات في مخابئها، يجب إبلاغ كل المعالجات الأخرى إلى أن البيانات الموجودة في مخابئها ليست صحيحة الآن.
- التسمية الإستعاريّة للعنوان الظاهري. يجب أن يتيح المعالج إدخال جدولي صفحات في نفس المعالجة للتخطيط إلى نفس إطار الصفحة. وغالباً ما يشارك نظام التشغيل صفحة مع معالجة مستعمل عن طريق تخطيط إدخال جدول صفحة ثانية.

تعتمد أجزاء معيّنة من برنامج إدارة VM على مزايا المعالج حيث يعمل نظام التشغيل. يجب تعديل أجزاء من برنامج إدارة الذاكرة المسردة في أعلى الصفحة التالية لكل منصّة عتاد توصل إليه.

- إدخالات جدول الصفحات. عندما يكون إدخال جدول صفحات صالحاً، يقسم المعالج 32 بت إلى حقول ويضبطها وفقاً لذلك. وعندما يكون إدخال جدول الصفحات غير صالح، يستعمل برنامج إدارة VM 31 بت الباقية عندما يختارها. يعتمد النسق الذي يختاره على وسائل الذاكرة الظاهريّة التي يوفّرها المعالج.

- حجم الصفحة. تستعمل المعالجات المختلفة أحجام صفحات مختلفة. يحدّد برنامج إدارة VM الذاكرة الظاهريّة على حدود 64 كيلوبايت التي تضمن إمكانيّة دعمه لأي حجم صفحة من 4 إلى 64 كيلوبايت. ولا تدعم أحجام الصفحة الأصغر من 4 كيلوبايت.
- الحماية الصفحيّة. الطريقة التي يتناول فيها برنامج إدارة VM حماية الصفحة العتاديّة لتنفيذ حماية صفحة براجمياتيّة إضافيّة تعتمد على العتاد.
- ترجمة العنوان الظاهري. اللوغاريتم الذي يستعمله برنامج إدارة VM لترجمة عنوان ظاهري إلى إدخال جدول صفحات يعتمد على العتاد.

4-6 بإختصار:

يستخدم برنامج إدارة VM في Windows NT نظام ذاكرة ظاهريّة معقّد. وهو يوفّر الوصول لكل معالجة إلى عدد كبير من العناوين الظاهريّة وحماية ذاكرة معالجة واحدة من ذاكرات الأخرى لكنّه يتيح للمعالجات مشاركة الذاكرة بفعاليّة وبدرجة تحكُّم كبيرة. ومع حقوق الوصول المناسبة، تستطيع معالجة أيضاً إدارة فسحة العنوان للمعالجات الأخرى وهي مزيّة تستخدمها الأنظمة الفرعيّة للمحيط. كذلك، تتوفّر قدرات متقدّمة، مثل الملفّات التخطيطيّة والقدرة على تحديد الذاكرة. يوفّر النظام الفرعي للمحيط Win 32 العديد من قدرات الذاكرة الظاهريّة NT للتطبيقات في روتينات 32-bit API.

يستخدم برنامج إدارة VM حماية الذاكرة الصفحيّة التي تزيد الحماية التي توفّرها المعالجات. وتستخدم مناطق الذاكرة المشاركة ككائنات، وبالتالي يتمّ التحكُّم بإستعمالها ومراقبتها بواسطة آليات الأمان التي تحمي كل الكائنات. إضافة لذلك، تستطيع معالجة إضافة حمايات تعتمد على الصفحة لإنتقاء أجزاء من الذاكرة المشاركة.

يعتمد إستعمال برنامج إدارة VM على طرق التقييم الكسول حيث أمكن لتجنُّب الأداء غير الضروري والعمليات المستغرقة للوقت ما لم تكن مطلوبة. وهذه واحدة من عدّة إستمثالات يستعملها برنامج إدارة VM لضمان الوصول السريع والكافي إلى الذاكرة.

يعالج الفصل التالي النواة NT، المركز الحقيقي للنظام Windows NT.

7

النواة

يصف مطوِّرو البرنامج التنفيذي NT مكوّن النواة العائد له على أنه «أسفل السلسلة الغذائيّة». وهذا المجاز صحيح على الأقل من ناحية واحدة. يتألف نظام التشغيل كأي جسم كبير من البراجيات، من طبقتين فوق بعضهما من الشيفرة. وتعتمد الطبقات الأعلى على الوظائف الأوليّة (لكن في هذه الحالة الأكثر قوّة) وبنيات البيانات المتوفّرة من قبَل الطبقات الأسفل. يشبه مجاز آخر النواة إلى صرّة عجلة. فهي مركز نظام التشغيل الذي يدور حوله كل شيء آخر.

تنفّذ النواة العمليّات الأكثر أساسيّة في Windows NT، حيث تحدّد كيفيّة إستعمال نظام التشغيل للمعالج أو المعالجات مع ضمان إستعمالها بتعقُّل. وهكذا، يعتمد نجاح كل نظام التشغيل على العمليّة الصحيحة والكافية للنواة.

مع هذا التحدّي، صمّم Dave Cutter مدير مجموعة تطوير Windows NT والمصمّم الرئيسي للنظام، وطبّق النواة NT.

لقد كان هدف Dave الرئيسي للنواة NT توفير قاعدة بمستوى منخفض من الآليات والوظائف الأوليّة المعرّفة بشكل جيّد لنظام التشغيل المتوقّع والتي تتيح للمكوّنات بمستوى أعلى للبرنامج التنفيذي NT تنفيذ ما يتوجّب عليها. وبإستعمال الوظائف الأوليّة للنواة، يستطيع البرنامج التنفيذي NT بناء خلاصات متنوّعة. ولا تحتاج لإستعمال التداخلات الجانبيّة والتأثيرات الجانبيّة غير الموجودة في مستند أو المناولة المباشرة للعتاد. تفصل النواة نفسها عن بقيّة البرنامج التنفيذي بإستخدام آليات نظام التشغيل وتجنُّب إتخاذ القرارات. وهي تترك كل القرارات إلى البرنامج التنفيذي NT.

وعن طريق توفير مجموعة وافرة من الآليات المحكومة المتناسقة، تمكّن النواة NT النظام Windows NT من النمو والتغيّر مع مرور الزمن لكن بطريقة منظمة متوقّعة. وقد ذكر Richard Rashid، خلال توضيحه عن لماذا قام هو وزملاؤه بإنشاء نظام التشغيل Mach (نموذج

مستضاف / ملقّم للنظام UNIX) «. . . لقد أصبحت نواة UNIX «مكبّ النفايات» لكل مزيّة أو وسيلة جديدة. . . . ولقد أصبحت الإستخلاصات ممزوجة والمعلومات ملخبطة». وهذه هي النتيجة الحتميّة لأنظمة التشغيل المستعملة بكثرة خلال مدة خدمتها على مدى عقدين. وقد إستهلكوا إلى أقصى حد ثم مدّدوا ثم عاودوا إستهلاكها إلى أقصى حد. غير أن النواة NT، بإعتمادها على الوظائف الأوليّة المدوّرة البسيطة وبرفض السياسات التي قد تصبح قديمة، تحاول حماية البرنامج التنفيذي NT من مصير «مكبّ النفايات» الحتمي.

1-7 نظرة شاملة:

إن فصل آليات نظام التشغيل عن سياساتها هو مبدأ مهمّ في Windows NT. وتنسب الآليات إلى الطريقة التي تنفّذ فيها المهام في نظام، الممثّلة باللوغاريتمات والشيفرة. تحدّد السياسات المهام الواجب تنفيذها وحتى أو حتى إذا وجب تنفيذ مهام معيّنة. تساعد شيفرة نظام التشغيل التي تفصل بشكل واضح بين الآليات والسياسات، على بقاء نظام التشغيل مرناً. فالسياسات يمكن أن تتغيّر بمرور الزمن دون أن تسبّب تموُّجاً في تغييرات كل النظام أو تغيُّر الآليات أيضاً.

يتواجد مبدأ فصل السياسات عن الآليات في عدّة مستويات في Windows NT. ففي المستوى الأعلى، ينشىء كل محيط فرعي لمحيط طبقة من سياسات نظام التشغيل التي تختلف عن تلك العائدة للأنظمة الفرعيّة الأخرى. وتحت الأنظمة الفرعيّة، ينشىء البرنامج التنفيذي NT طبقة أساسيّة أخرى من السياسات التي تستوعب كل الأنظمة الفرعيّة. وفي الطبقة السفلى من نظام التشغيل، تتجنّب النواة إتخاذ القرارات. لكنّها وعِوَضاً عن ذلك، تستخدم كطبقة بين بقيّة نظام التشغيل والمعالج. ويؤدّي تمرير كل العمليات المتعلقة بالمعالج عبر النواة إلى نقليّة وتوقُّعيّة أفضل. يسلّط البرنامج التنفيذي تحكُّماً محدوداً فقط على هذه العمليّات عن طريق إستدعاء وظائف النواة.

إضافة إلى الوظائف التي توفّرها إلى البرنامج التنفيذي NT، تنفّذ النواة أربع مهام رئيسيّة:

- جدولة الشعَب للتنفيذ.
- نقل التحكُّم إلى روتينات برنامج المناولة عند حصول مقاطعات وإستثناءات.
- تنفيذ مزامنة متعدّدة المعالجات بمستوى منخفض.
- إستخدام إجراءات إستعادة النظام بعد حصول إنقطاع في الطاقة الكهربائيّة.

تختلف النواة عن بقيّة البرنامج التنفيذي في عدّة طرُق. وبعكس الأجزاء الأخرى من

البرنامج التنفيذي، لا تخرج النواة أبداً من الذاكرة. ويشكل مشابه، ورغم إمكانيّة مقاطعتها لتنفيذ روتين خدمة مقاطعة (راجع الفصل الثامن «نظام الدخل / الخرج») فإن تنفيذها ليس شفعياً. بمعنى آخر، تتوقّف المهام المتعدّدة لفترات زمنيّة قصيرة خلال إشتغال النواة. تشتغل النواة دائماً في نمط النواة، وهو نمط المعالج المفضّل في Windows NT ومصمّم ليكون صغيراً ومتراصًا ونقّالاً قدر ما يتيح به الأداء والإختلافات في تصاميم المعالج. لقد كُتبت شيفرة النواة باللغة C مع حجز شيفرة assembly لتلك المهام التي تتطلّب شيفرة سريعة أو تلك التي تعتمد على قدرات المعالج.

خارج النواة، يمثّل البرنامج التنفيذي الشعَب والموارد المشاركة الأخرى ككائنات. تتطلّب هذه الكائنات بعض الكلفات الإضافيّة للسياسات، مثل مقابض الكائن لمناولتها والتدقيقات الأمنيّة لحمايتها وحصص الموارد الواجب حسمها عند إنشائها والطرق العاديّة لتحديد الذاكرة وإزالتها لحفظها. تزال الكلفة الإضافيّة في النواة، التي تستخدم مجموعة من الكائنات الأبسط التي تسمى كائنات النواة والتي تساعد النواة على التحكّم بالمعالجة المركزيّة ودعم إنشاء كائنات تنفيذيّة. تحتوي معظم الكائنات بمستوى تنفيذي كل كائن نواة واحد أو أكثر يتضمّن الصفات القويّة المعرّفة من قبل النواة.

تنشىء إحدى مجموعات كائنات النواة، التي تسمى كائنات التحكُم، ألسنيّة للتحكّم بوظائف نظام التشغيل المتنوّعة. تتضمّن هذه المجموعة كائن معالجة النواة، كائن إستدعاء إجراء لا تزامني (APC) وكائن إستدعاء إجراء مؤجّل (DPC) وعدّة كائنات مستعملة من قبل نظام الدخل / الخرج بما فيها كائن المقاطعة وكائن إبلاغ الطاقة وكائن حالة الطاقة. تتضمّن مجموعة أخرى من كائنات النواة والتي تسمى كائنات التوزيع، قدرات المزامنة والتعديل والتأثير على جدولة الشعَب. تتضمّن كائنات التوزيع شعبة النواة، وخافت النواة، وظافر النواة، وحدث النواة، وزوج أحداث النواة، والإعلام الإشاري للنواة، وموقّت النواة. يستعمل البرنامج التنفيذي وظائف النواة لإنشاء حالات آنيّة لكائنات النواة ولمناولتها ولإنشاء الكائنات الأكثر تعقيداً التي توفّرها لنمط المستعمل. تشرح كائنات النواة بتفصيل أكبر في هذا الفصل وفي الفصل التالي.

يمثّل وصف تصميم النواة مشكلة دون حلّ. فلا توجد نقطة بداية معيّنة لأن كل جزء من النواة يعتمد على الأجزاء الأخرى. لذلك، تعرض المواضيع في هذا الفصل وفقاً لأهميتها لوظائف نظام التشغيل. يشرح أولاً جدولة الشعَب وتوزيعها، يتبع ذلك مناولة المقاطعة والإستثناء. يوصف بعد ذلك مزامنة المعالجات المتعدّدة ويتبع بالإستعادة عند إنقطاع الطاقة الكهربائيّة وهي موضوع يتعلّق عن كثب بموضوع الفصل الثامن ــ نظام الدخل / الخرج.

2-7 جدولة الشعَب وتوزيعها:

الشعبة هي وحدة مستقلّة قابلة للتنفيذ تشتغل في فسحة عنوان معالجة بإستعمال الموارد المخصّصة للمعالجة. إحدى وظائف النواة NT هي تعقُّب الشعَب الجاهزة للتنفيذ ولإنتقاء ترتيب تشغيلها، وهي مهمّة تُعرَف بإسم جدولة الشعَب. وعندما تكون الظروف مؤاتية، تنتقي النواة شعبة جديدة لتشغيلها وتنفذ عملية تبديل سياقي عليها. والتبديل السياقي هو إجراء حفظ حالة الماكنة المتطايرة وبدء تنفيذ شعبة جديدة. تسمى الوحدة التي تنفّذ هذه المهام موزّع النواة.

يبدأ هذا القسم ببعض المعلومات العامة حول كيفيّة عرض النواة للشعَب والمعالجات تتبع بشرح حول مهام الموزّع: الجدولة والتبديل السياقي.

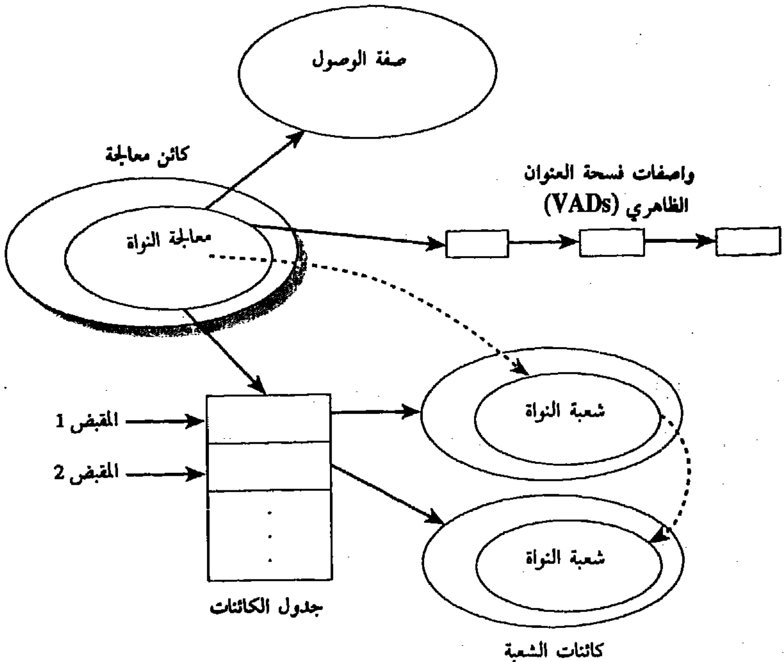
1-2-7 معالجة النواة وكائنات الشعبة:

إن وظيفة الموزّع هي ضمان أنه من بين كل الشعَب التي تنتظر التنفيذ، ينفّذ المعالج دائماً أكثر الشعَب أهميّة. وعند حصول أحداث نظام تغيّر حالة بعض الشعَب يراجع الموزّع حالة الشعَب المنتظرة وينفّذ تبديل سياقي إلى شعبة جديدة عند ضرورة التغيير.

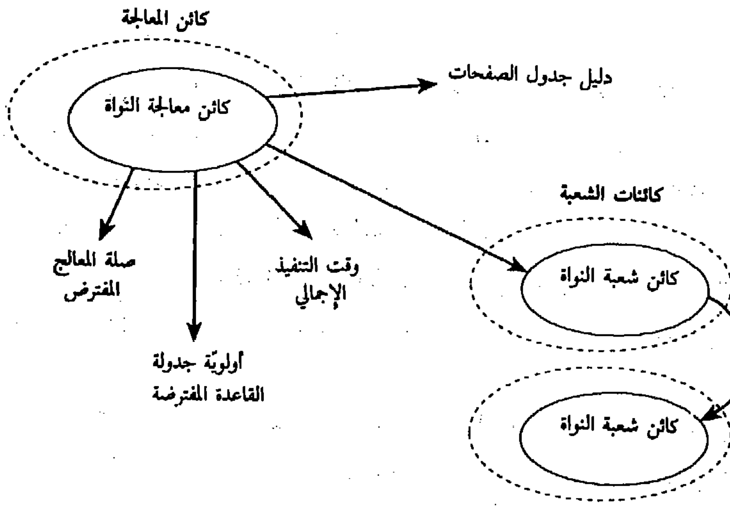
رغم أن الموزّع يتناول الشعَب، إلا أنه لا يشارك نفس مشهد الشعَب كما تفعل البرامج في نمط المستعمل أو كما تفعل بقيّة نظام التشغيل. تعمل النواة مع النموذج المشذّب لكائن الشعبة، الذي يسمى كائن شعبة النواة.

يتواجد كائن شعبة النواة ضمن كائن شعبة تنفيذيّة ويعرض فقط المعلومات التي تحتاجها النواة لتوزيع الشعبة للتنفيذ. بشكل مشابه، تستخدم النواة نموذجاً مصغّراً من كائن معالجة، يسمّى كائن معالجة النواة. يوضح الشكل (7-1) العلاقة بين كائنات معالجة النواة وشعبة النواة ونظيرها التنفيذي ذات المستوى الأعلى.

كما يظهر في الشكل (7-2) على الصفحة التالية، يحتوي كائن معالجة النواة مؤشّراً إلى لائحة شعَب النواة. (لا تعرف النواة بوجود المقابض، لذلك فإنها تتجاوز جدول الكائنات)، كذلك يشير كائن معالجة النواة إلى دليل جدول صفحات المعالجة (المستعمل لتعقُّب فسحة العنوان الظاهري للمعالجة) والوقت الإجمالي الذي نفّذته شعَب المعالجة وأولويّة جدولة القاعدة المفترضة للمعالجة والمجموعة المفترضة للمعالجات حيث تستطيع الشعَب الإشتغال (تسمى صلة المعالج). تتحكّم النواة بالمعلومات المخزّنة في كائن معالجة النواة. يستطيع بقيّة البرنامج التنفيذي قراءة أو تعديل المعلومات فقط عن طريق إستدعاء وظيفة نواة.



الشكل (7-1)
كائن معالجة النواة وكائنات شعبة النواة



الشكل (7-2)
كائن معالجة النواة

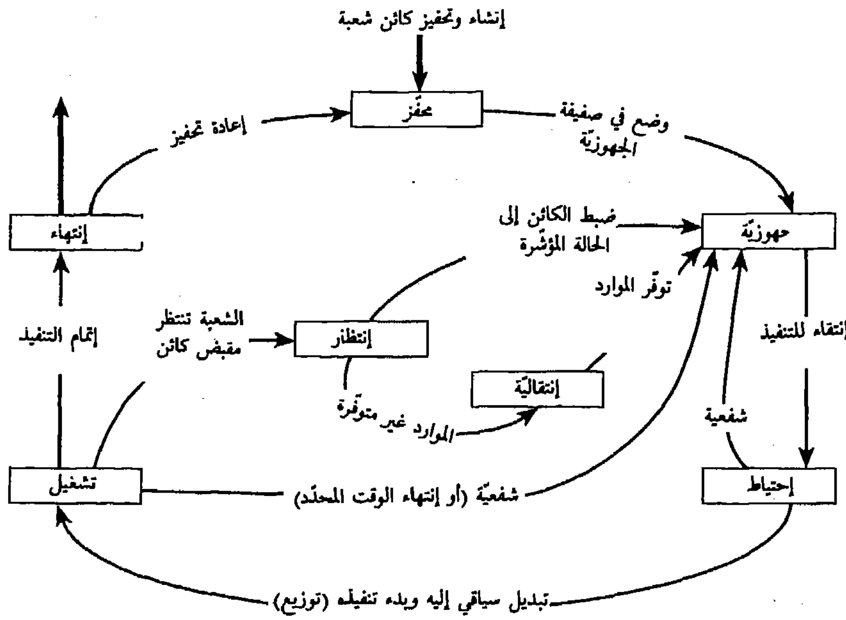
إن كائن شعبة النواة أكثر تعقيداً من كائن معالجة النواة. وهو يحتوي بعض المعلومات الواضحة مثل صلة معالج الشعبة (مجموعة فرعيّة غير صحيحة من ضوابط المعالجة المفترضة) وكميّة الوقت الإجماليّة التي نفّذتها الشعبة. وهو يشمل أيضاً أولويّة جدولة قاعدة الشعبة (التي يمكن أن تختلف عن أولويّة جدولة القاعدة المفترضة المعيّنة لمعالجتها) والأولويّة الحاليّة للشعبة مجموعة من البيانات المهمّة بشكل خاص التي يحتويها كائن شعبة النواة هي حالة شعبة الموزّع. يمكن أن تكون الشعبة في أي حالة من ستّ حالات في أي وقت، وواحدة منها فقط تجعل الشعبة صالحة للتنفيذ. توضّح حالات شعَب الموزّع في الشكل (7-3) على الصفحة 219.

تبدأ دورة حياة الشعبة عندما ينشىء برنامج شعبة جديدة. ويرشح الطلب إلى البرنامج التنفيذي NT، حيث يحدّد برنامج إدارة المعالجة فسحة لكائن شعبة، ويستدعي النواة لتحفيز كائن شعبة النواة الموجودة ضمنها. بعد تحفيزها، تتابع الشعبة عبر الحالات التالية:

- الجهوزيّة: عند البحث عن شعبة لتنفيذها، يعتبر الموزّع فقط مجموعة الشعَب في حالة الجهوزيّة. وهذه الشعَب تنتظر التنفيذ.
- الإحتياط: تمَّ إنتقاء شعبة في حالة الإحتياط لتشتغل في المرّة التالية على معالج معيّن. وعندما تتواجد ظروف مؤاتية، ينفّذ الموزّع تبديل سياقي على هذه الشعبة. يمكن لشعبة واحدة فقط أن تكون في حالة الإحتياط لكل معالج على النظام.
- التشغيل: عندما ينفّذ الموزّع تبديل سياقي إلى شعبة، تدخل الشعبة حالة التشغيل وتنفّذ. يستمرُّ تنفيذ الشعبة إلى أن إما تشفعها النواة لتشغّل شعبة بأولويّة أعلى، أو إنتهاء وقتها المحدّد أو إنهائها أو دخولها طوعياً حالة الإنتظار.
- الإنتظار: تستطيع الشعبة دخول حالة الإنتظار في عدّة طرُق: تستطيع الشعبة إنتظار كائن طوعياً لمزامنة تنفيذه، ويستطيع نظام التشغيل (نظام الدخل / الخرج، مثلاً) الإنتظار نيابة عن الشعبة، أو يستطيع نظام فرعي لمحيط توجيه الشعبة لتعلّق نفسها. عند إنتهاء إنتظار الشعبة، تعدّد الشعبة إلى حالة الجهوزيّة لمعاودة جدولتها.
- الإنتقاليّة: تدخل الشعبة الحالة الإنتقاليّة إذا كانت جاهزة للتنفيذ لكن الموارد التي تحتاجها غير متوفّرة. فمثلاً، يمكن إخراج تكديس نواة الشعبة من الذاكرة. وبعد توفّر مواردها، تدخل الشعبة حالة الجهوزيّة.
- الإنهاء: عندما تنتهي شعبة من التنفيذ، فإنها تدخل حالة الإنهاء ــ بعد إنهائها يمكن حذف كائن شعبة أو لا يمكن ذلك. (يضع برنامج إدارة الكائنات سياسة تتعلّق بمتى تُحذف الكائنات). وإذا كان البرنامج التنفيذي يحتوي على مؤشّر إلى كائن الشعبة، فإنه يستطيع إعادة تحفيز كائن الشعبة وإستعماله مجدّداً.

تحتاج حالة الإنتظار لمزيد من الشرح. تكون الشعبة في حالة الإنتظار عندما تكون تنتظر ضبط كائن أو مجموعة كائنات إلى حالة مؤشّرة. وكما شرح في الفصل الرابع، «المعالجات والشعَب»، تكون الكائنات التنفيذيّة التي تدعم المزامنة في حالة من حالتين: مؤشّرة أو غير مؤشّرة. تبقى الكائنات في الحالة غير المؤشّرة إلى أن يحصل حدث مهمّ. فمثلاً، تضبط شعبة إلى الحالة المؤشّرة عند إنهائها. وتفلت شعَب المستعمل التي تنتظر مقبض الشعبة المنتهية وتستطيع مواصلة التنفيذ. وبشكل مشابه يضبط كائن ملفّ إلى الحالة المؤشّرة عند إتمام عمليّة دخل / خرج مطلوبة. وتفلت الشعبة التي تنتظر مقبض الملفّ من حالة الإنتظار وتستطيع مواصلة التنفيذ.

إن النواة هي التي تستخدم ألسنيّة الإنتظار والتأشير في Windows NT (وهي غير إشارات POSIX التي تشبه إستثناءات NT). يتضمّن كل كائن مزامنة مرئي لنمط المستعمل كائناً واحداً أو عدّة كائنات توزيع النواة. فمثلاً، يحتوي كائن الشعبة شعبة نواة ويحتوي كائن الحدث حدث نواة ويحتوي كائن الملفّ ملفّ نواة. تكون النواة مسؤولة عن ضبط كائنات التوزيع إلى الحالة



الشكل (3-7)
حالات الشعبة

المؤشّرة وفقاً للقوانين المعرّفة بشكل جيّد. وعندما تقوم بذلك، فإنها تفلت الشعَب التي كانت تنتظر هذه الكائنات عن طريق تنفيذ حالة التوزيع من الإنتظار إلى الجهوزيّة. وهذه بدورها تحثُّ الموزّع على تحفيز جدولة الشعَب وهو موضوع القسم التالي. يعاود هذا الفصل التحدُّث عن موضوع كائنات توزيع النواة والمزامنة.

2-2-7 جدولة الأولويّات

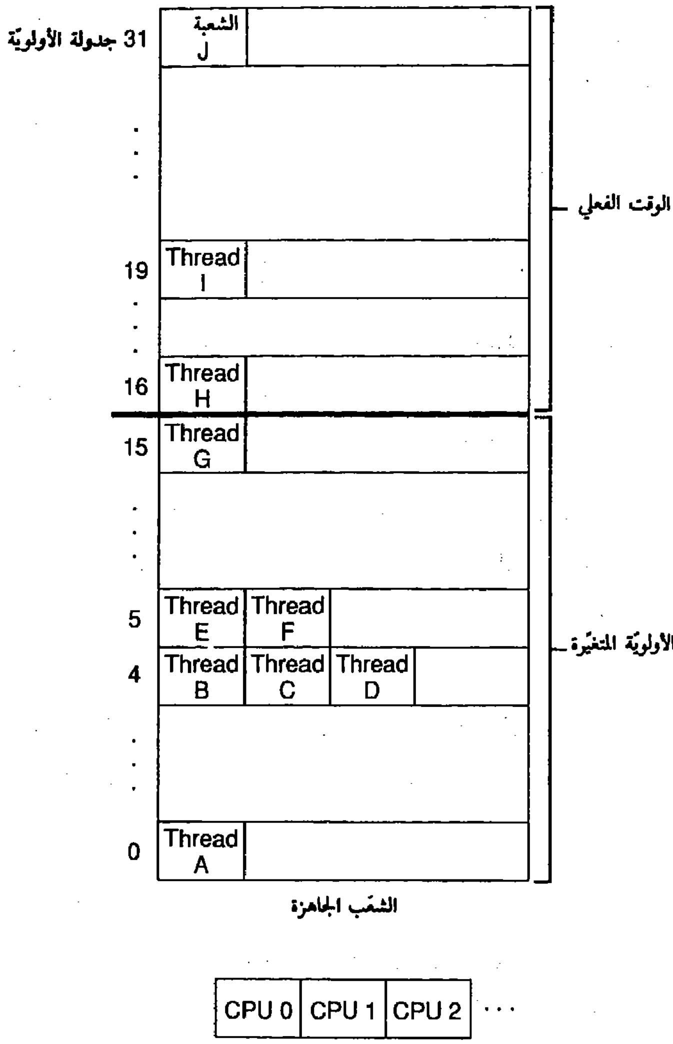
يستعمل موزّع النواة مخطّط أولويّة لتحديد ترتيب تنفيذ الشعَب، حيث يجدول الشعَب بأولويّة أعلى قبل تلك بأولويّة أدنى. وتوقّف النواة وحتى تشفع تنفيذ شعبة إذا أصبحت شعبة بأولويّة أعلى جاهزة للتنفيذ.

مبدئياً، تحصل الشعبة على أولويّتها من المعالجة التي أنشأت فيها. فمثلاً، عندما ينشىء نظام فرعي لمحيط معالجة، فإنّه يعيّن أولويّة مرجعيّة مفترضة للمعالجة (نظام مفترض أو رقم مضبط من قبل مدير النظام). تتأصّل الشعبة الأولويّة المرجعيّة وتستطيع تعديلها لتكون أعلى أو أدنى بقليل. هذه هي أولويّة الجدولة حيث تبدأ الشعبة بالتنفيذ. وقد تتغيّر أولويّة الشعبة عن هذه القاعدة عند التنفيذ.

لإتخاذ قرارات جدولة الشعَب، تحتفظ النواة بمجموعة من بنيات البيانات المعروفة جماعياً بإسم قاعدة بيانات الموزّع. تتعقّب قاعدة بيانات الموزّع الشعَب التي تنتظر التنفيذ والمعالجات التي تنفّذ الشعَب. تسمى أكثر البنيات أهميّة في قاعدة بيانات الموزّع، صفيفة جهوزيّة الموزّع. وهذه الصفيفة هي في الواقع سلسلة من الصفيفات، بحيث يكون هناك صفيفة واحدة لكل أولويّة جدولة. تحتوي الصفيفات المبيّنة في الشكل (7-4) على الصفحة التالية، الشعَب الموجودة في حالة الجهوزيّة بإنتظار جدولتها للتنفيذ.

وكما يوضح الشكل، يدعم البرنامج التنفيذي NT,32 مستوى أولويّة، مقسّمة إلى فئتين؛ الوقت الفعلي والأولويّة المتغيّرة. إن شعَب الوقت الفعلي ذات الأولويّات 16 إلى 31، هي شعَب بأولويّة عالية مستعملة من قبَل البرامج الدقيقة زمنياً ـــ مثل مراقبة أجهزة القياس ـــ الذي يتطلّب إنتباهاً من المعالج.

عندما يعاود الموزّع جدولة معالج، فإنه يبدأ عند الصفيفة بأعلى أولويّة وينزل إلى أن يجد شعبة وبالتالي فإنه يجدول كل شعَب الوقت الفعلي قبل جدولة أية شعَب بأولويّة متغيّرة. تقع معظم الشعَب في النظام في فئة الأولويّة المتغيّرة، مع إمتداد الأولويّات من 1 إلى 15 (تحجز الأولويّة 0 لإستعمال النظام). تسمّى هذه الشعَب الأولويّة المتغيّرة لأن الموزّع يعدّل أولويّتها عند



الشكل (4-7)
صفيفة جهوزيّة الموزّع

التنفيذ لإستمثال وقت إستجابة النظام. فمثلاً، ولأن النظام Windows NT نظام متعدّد المهام شفعي، يقاطع الموزّع شعبة بعد تنفيذها لحصّة الوقت المحدّد الكامل. وإذا كانت الشعبة المقاطعة هي شعبة بأولويّة متغيّرة، يخفّض الموزّع أولويّة الشعبة. وهكذا، تختفي تدريجياً الشعبة المربوطة حسابياً (إلى الأولويّة المرجعيّة).

من الناحية المقابلة، يرفع الموزّع أولويّة شعبة بعد إفلاتها من عمليّة الإنتظار. تحدّد عادة الشيفرة التنفيذيّة خارج النواة حجم تعزيز أولويّة الشعبة، لكن حجم التعزيز يتبع نموذجاً وفقاً لما كانت تنتظره الشعبة. فمثلاً، تستلم شعبة تنتظر دخل لوحة مفاتيح، تعزيزاً أكبر من تلك التي تنتظر إتمام دخل / خرج قرص. إجمالاً، تشتغل الشعَب التفاعليّة عند أولويّة متغيّرة مرتفعة والشعَب المربوطة بالدخل / الخرج عند أولويّة متوسطة والشعبة المربوطة حسابياً عند أولويّة منخفضة. (يمكن تعزيز أولويّة شعبة بأولويّة متغيّرة إلى فئة الوقت الفعلي).

كذلك تلعب صلة معالج الشعبة دوراً في تقرير ترتيب تنفيذ الشعَب، تنتقي النواة شعبة وفقاً لأولويّتها ثم تدقّق بالمعالج حيث تستطيع الشعبة الإشتغال. فإذا كانت صلة معالج الشعبة لا تتيح لها التشغيل على أي معالج متغيّر، عندئذٍ تنتقي النواة الشعبة بالأولويّة الأعلى التالية.

عند عدم حصول أي شيء في النظام، تزوّد النواة شعبة واحدة (لكل معالج) يمكنها التنفيذ دائماً. تسمى هذه الشعَب، الشعَب المتوقّفة ويعاملها الموزّع وكأن أولويّتها دون تلك العائدة للشعَب الأخرى. عادة تدور الشعبة المتوقّفة في حلقات حيث تدقّق لجهة دخول شعبة أخرى حالة إحتياط للتنفيذ على معالجها.

وعندما تكتشف شعبة، تحفز الشعبة المتوقّفة تبديل سياقي إلى تلك الشعبة. كذلك تدقّق الشعبة المتوقّفة بوجود أية روتينات إستدعاء إجراء مؤجّل (DPC) يجب تنفيذها. (يتمّ وصف روتينات DPC في القسم 7-3-2-3).

3-2-7 التبديل السياقي:

بعد تنفيذ شعبة لكميّة الوقت الكاملة، تشفعها النواة وتعيد جدولة المعالج. لكن نفاذ كميّة الوقت ليست العامل الوحيد الذي يحفز جدولة الشعبة. فالجدولة مدارة بحدث تحفّز عندما لا تتمكّن الشعبة الشغّالة من الإستمرار أو عندما تتغيّر حالة شعبة ولا تعود شعبة التنفيذ بأولويّة أعلى ــ توجد بعض الأمثلة عن الظروف التي تؤدّي إلى إعادة الجدولة أدناه:

- عندما تصبح شعبة جاهزة للتنفيذ ــ مثلاً، شعبة محفزة حديثاً أو أُطلقَت للتوّ من حالة الإنتظار.
- عندما تنتهي كميّة وقت الشعبة، عندما تنتهي أو عندما تدخل حالة الإنتظار.
- عندما يغيّر الموزّع أو البرنامج التنفيذي (ربما بناءًا لطلب برنامج تطبيق) أولويّة شعبة.
- عندما يتغيّر البرنامج التنفيذي أو البرنامج التطبيقي صلة المعالج لشعبة شغّالة.

إن الهدف من إعادة الجدولة هو إنتقاء الشعبة للتنفيذ تالياً على معالج معيّن ووضعها في حالة الإحتياط. لكن إيجاد الشعبة ليس كافياً. فعلى الموزّع أن يبدأ أيضاً التنفيذ.

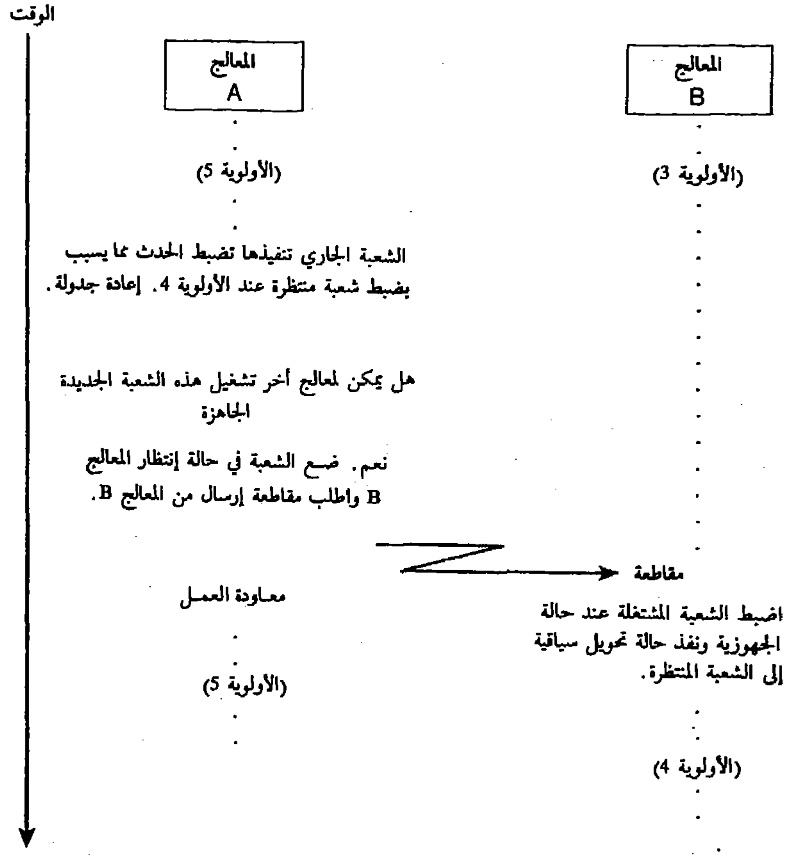
إذا أُنهيت الشعبة التي كانت تشتغل أو تعذّرَ مواصلة تنفيذها، ينفّذ الموزّع تبديل سياقي إلى شعبة جديدة. وإلا، يجب على الموزّع القيام بأشياء إضافيّة. فمثلاً، إذا أصبحت شعبة وقت فعلي بأولويّة مرتفعة، جاهزة للتنفيذ لكن ما تزال شعبة بأولويّة أدنى شغّالة، على الموزّع أن يشفع الشعبة المنفّذة. ولشفعة شعبة، يطلب الموزّع مقاطعة البراجيات لتحفيز تبديل سياقي، كما يبيّن في الشكل (7-5) على الصفحة التالية.

عند إعادة جدولة الشعَب، تستعمل النواة قاعدة بيانات الموزّع ليحدّد بسرعة المعالجات المشغولة والمتوقّفة (التي تعمل على الشعبة المتوقّفة) وأولويّة الشعبة التي ينفّذها كل معالج. في هذا المثال، تحدّد النواة (الشغّالة على المعالج A) أن المعالج B يشغّل شعبة بأولويّة أدنى من تلك العائدة لشعبة جاهزة حديثاً. تطلب النواة مقاطعة التوزيع لشفعة الشعبة الشغّالة على المعالج B تستجيب النواة التي تشتغل على المعالج B للمقاطعة عن طريق إعادة الجدولة، أي، تحفيز تبديل سياقي من شعبة الأولويّة 3 التي تشغلها إلى الشعبة الجديدة. يُعاد ضبط الشعبة الشفعيّة إلى الحالة الجاهزة وترجع إلى صفيفة جهوزيّة الموزّع لجدولتها لاحقاً، ربما على معالج مختلف. ورغم أنه يتمّ وضع معظم الشعب في طرف الصفيفة وفقاً لأولويّتها، توضع الشعبة الشفعيّة في الوضعيّة الأولى من الصفيفة.

يغيّر سياق شعبة وإجراء التبديل السياقي وفقاً لتصميم المعالج يتطلّب التبديل السياقي العادي حفظ البيانات التالية وإعادة تحميلها:

- عدّاد البرنامج.
- مسجّل حالة المعالج.
- محتويات المسجّل الأخرى.
- مؤشّرات تكديس المستعمل والنواة.
- مؤشّر إلى فسحة العنوان حيث تشتغل الشعبة (دليل جدول صفحات المعالجة).

لتنفيذ تبديل سياقي، تحفظ النواة هذه المعلومات بدفعها على تكديس نمط النواة للشعبة الحاليّة وتحديث مؤشّر التكديس. تحمّل النواة سياق الشعَب الجديدة وإذا كانت الشعبة موجودة في معالجة مختلفة، فإنها تحمّل عنوان دليل جدول الصفحات لكي يتوفّر فسحة عنوانها. بعد أعمال التنظيف التي تقوم بها النواة، يمرّر التحكّم إلى عدّاد البرنامج المستعاد للشعبة الجديدة وتبدأ الشعَب بالتنفيذ.



الشكل (7-5)
إرسال الشُعب في نظام متعدد المعالجات

من الجدير هنا معاودة ذكر أنه رغم إمكانيّة شفعة شعبة تشغّل شيفرة البرنامج التنفيذي NT من قبَل شعبة بأولويّة أعلى، لا يمكن شفعة شعبة تشغّل شيفرة النواة. فعندما تشتغل النواة، فإنها تشتغل بفاعليّة عند أولويّة أعلى من أية شعبة أخرى في النظام. (في الواقع، تقوم النواة بإزالة تمكين توزيع الشعبة). ورغم تعذُّر شفعها، يمكن مقاطعة معظم النواة بواسطة المقاطعات بأولويّة أعلى، وهو موضوع سيُشرح لاحقاً في هذا الفصل.

3-7 مناولة المقاطعة والإستثناء:

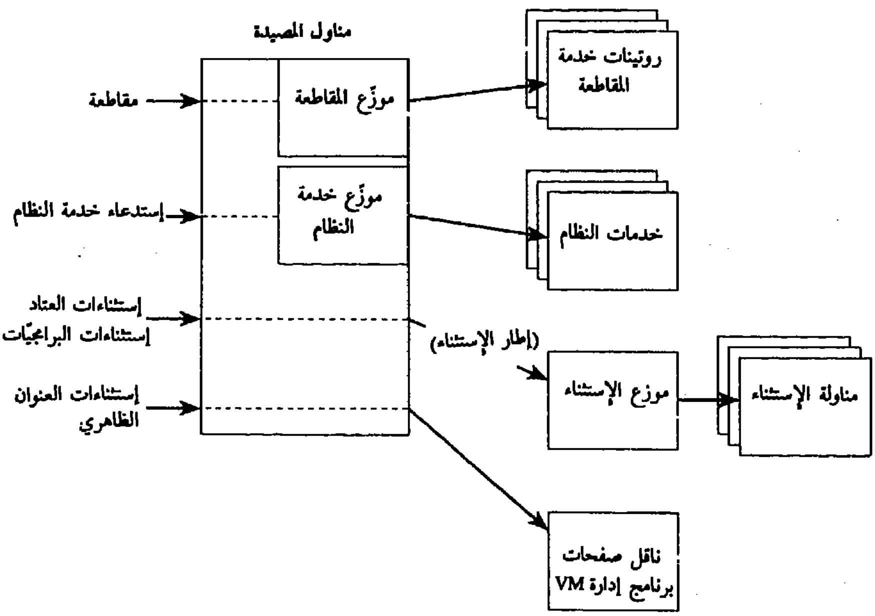
المقاطعات والإستثناءات هي ظروف نظام تشغيل تحوّل المعالج إلى شيفرة خارج الإنسياب العادي للتحكّم. ويمكن أن تكتشف بواسطة العتاد أو البراجيّات. وعند كشف مقاطعة

أو إستثناء، يوقف المعالج ما يقوم به وينقل التحكُّم (يوزِّع) إلى موقع مستقلّ في الذاكرة، إلى عنوان الشيفرة التي تعالج الظرف. تسمى هذه الشيفرة في NT، مناول المصيدة.

تفرّق النواة NT بين المقاطعات والإستثناءات بالطريقة التالية. فالمقاطعة هي حدث لاتزامني يحصل في أي وقت وغير متعلّق بما ينفّذه المعالج. وتولّد المقاطعات مبدئياً من قبل أجهزة الدخل / الخرج وإقفال المعالج أو المؤقّتات ويمكن تمكينها (تشغيلها) أو إلغاء تمكينها (توقيفها).

من الناحية المقابلة، فالإستثناء هو حالة تزامنيّة تنتج عن تنفيذ تعليمة معيّنة. يمكن إعادة توليد الإستثناءات بتشغيل نفس البرنامج مع نفس البيانات تحت نفس الظروف. تشمل أمثلة الإستثناءات مخالفات الوصول إلى الذاكرة وبعض تعليمات مزيل العِلل المعيّنة وأخطاء القسمة إلى صفر. كذلك تعتبر النواة NT إستدعاءات خدمة النظام كإستثناءات (رغم أنها مصايد نظام من الناحية التقنيّة).

يركّز الشرح التالي على مناولة المقاطعة والإستثناء كما تستخدم على المعالج MIPS R4000. وقد يؤدّي تطبيق Windows NT على التصاميم الأخرى إلى عملها بطريقة مختلفة نتيجة التغيير في الدعم المزوّد من قبَل المعالج.



الشكل (7-6)
توزيع المقاطعات والإستثناء

7-3-1 مناول المصيدة:

ينسب التعبير (مصيدة) إلى آلية معالج لإلتقاط شعبة تنفيذ عند حصول إستثناء أو مقاطعة والتبديل من نمط المستعمل إلى نمط النواة ونقل التحكُّم إلى موقع ثابت في نظام التشغيل. وفي Windows NT، ينقل المعالج التحكُّم إلى مناول مصيدة النواة NT وهي وحدة تعمل كلوحة مفاتيح عن طريق ضبط الإستثناءات والمقاطعات المكتشفة من قبَل المعالج في حقول ونقل التحكُّم إلى الشيفرة التي تتناول الظرف.

يمكن توليد الإستثناءات والمقاطعات إما بواسطة العتاد أو بالبراجيات. فمثلاً، ينتج إستثناء خطأ ناقل عمومي عن مشكلة عتاد، بينما ينتج إستثناء القسمة على صفر من قبل علّة براجيّات. وبشكل مشابه، يستطيع جهاز دخل / خرج توليد مقاطعة أو تستطيع النواة نفسها إصدار مقاطعة براجيّات. يوضح الشكل (7-6) بعض الظروف التي تنشّط مناول المصيدة والوحدات التي يستدعيها مناول المصيدة لخدمتها.

عند تحفيزه، يلغي مناول المصيدة تمكين المقاطعات بإيجاز خلال تسجيله حالة الماكنة (المعلومات التي تسمح في حال حصول مقاطعة أو إستثناء). وهو ينشىء إطار مصيدة حيث تخزّن حالة التنفيذ للشعبة المقاطعة. تتيح هذه المعلومات للنواة معاودة تنفيذ الشعبة بعد مناولة المقاطعة أو الإستثناء. وعادة يكون إطار المصيدة عبارة عن مجموعة فرعيّة من السياق الكامل لشعبة. وكما ذكر في القسم السابق، يتغيّر سياق شعبة تصميم المعالج.

يحلّ مناول المصيدة بعض المشاكل بنفسه، مثل بعض إستثناءات العنوان الظاهري، لكن في معظم الحالات، فإنّه يحدّد الظرف الذي يحصل وينقل التحكُّم إلى النواة الأخرى أو الوحدات التنفيذيّة. مثلاً، إذا كان الظرف مقاطعة جهاز، تنقل النواة التحكُّم إلى روتين خدمة المقاطعة (ISR) المتوفّر من قبل مسيق الجهاز للجهاز المقاطع. وإذا نتج الظرف عن إستدعاء إلى خدمة نظام، ينقل مناول المصيدة التحكُّم إلى شيفرة خدمة النظام في البرنامج التنفيذي NT. وتضبط الإستثناءات المتبقّية في حقول بواسطة موزّع الإستثناء الخاص بالنواة. تصف الأقسام التالية توزيع المقاطعة وتوزيع خدمة النظام والمناولة الإستثنائيّة بتفصيل أكبر.

7-3-2 توزيع المقاطعة:

تصدر عادة المقاطعات الناشئة عن العتاد من أجهزة الدخل / الخرج التي يجب أن تبلغ المعالج عندما تحتاج للخدمة. وتتيح الأجهزة المدارة بالمقاطعة لنظام التشغيل الإستعمال الأقصى للمعالج عن طريق تراكب المعالجة المركزيّة مع عمليات الدخل / الخرج. فالمعالج يبدأ نقل الدخل / الخرج إلى جهاز ومنه ثم ينفّذ الشعَب الأخرى خلال إتمام الجهاز عمليّة النقل. وعند

إنتهاء الجهاز من ذلك، فإنه يقاطع المعالج للحصول على خدمة. إنّ أجهزة التأشير والطابعات ولوحات المفاتيح وسواقات الأقراص وبطاقات الشبكة هي أجهزة مدارة بالمقاطعة بشكل عام.

تستطيع براجيّات النظام أيضاً توليد المقاطعات. فمثلاً، تستطيع النواة إصدار مقاطعة براجيّات لتحفيز توزيع الشعبة ولتنفصل لاتزامنياً إلى تنفيذ شعبة. وتستطيع النواة إلغاء تمكين المقاطعات بحيث لا يستلمها المعالج، لكنه يستلمها على فترات غير متكرّرة ــ عند اللحظات الحرجة خلال معالجة مقاطعة أو توزيع إستثناء، على سبيل المثال.

تستجيب وحدة فرعيّة لمناول مصيدة النواة، تسمى موزّع المقاطعة، للمقاطعات. وهي تحدّد مصدر المقاطعة وتنقل التحكّم إما إلى روتين خارجي الذي يتناول المقاطعة، يسمّى روتين خدمة المقاطعة (ISR) أو إلى روتين نواة داخلي يستجيب للمقاطعة. وتزوّد مسيقات الأجهزة روتينات ISR لخدمة مقاطعات الجهاز وتوفّر النواة روتينات مناولة المقاطعة للأنواع الأخرى من المقاطعات.

يبدأ القسم الفرعي التالي بوصف أنواع المقاطعات التي تدعمها النواة NT. ويتبع ذلك تفاصيل إضافيّة حول معالجة المقاطعة، بما فيها شرح موجز حول الطريقة التي تتفاعل فيها مسيقات الأجهزة مع النواة. ويصف القسم الفرعي الأخير مقاطعات البراجيّات التي تتعرّف إليها النواة وكائنات النواة المستعملة لتطبيقها.

1-2-3-7 أنواع المقاطعات والأولويّات:

تستطيع المعالجات المختلفة التعرُّف إلى أعداد وأنواع مختلفة من المقاطعات. فموزّع المقاطعة يخطّط مستويات مقاطعة العتاد على مجموعة قياسيّة من مستويات طلب المقاطعة (IRQLs) المعروفة من قبل نظام التشغيل.

تصنّف مستويات IRQL المقاطعات وفقاً لأولويّتها. وتختلف أولويّات IRQL عن أولويّات الجدولة الموصوفة سابقاً. فأولويّة الجدولة هي صفة شعبة بينما IRQL هي صفة مصدر مقاطعة مثل لوحة المفاتيح أو الماوس. إضافة لذلك يحتوي كل معالج على ضبط IRQL أو تخفيضه للمعالج حيث تشتغل لحجب المقاطعات بمستوى أدنى أو إلغاء حجبها.

تعرّف النواة مجموعة من مستويات IRQL النقّالة التي تستطيع زيادتها إذا إتّصف المعالج بمزايا مقاطعة خاصة (ساعة ثانية، مثلاً). وتخدم المقاطعات وفقاً لأولويّتها، وتشفع المقاطعة بأولويّة أعلى خدمة مقاطعة بأولويّة أدنى. تبينّ مستويات IRQL النقّالة في الجدول (7-1) من الأولويّة الأعلى إلى الأدنى.

تحجز مستويات IRQL من المستوى الأعلى نزولاً إلى مستوى الجهاز 1 لمقاطعات العتاد وتكون مقاطعات مستوى التوزيع / DPC ومستوى APC مقاطعات براجيّات تصدرها النواة. وإن مستوى IRQL المنخفض ليس مستوى مقاطعة بتاتاً وإنما الضبط حيث تنفذ الشعبة العاديّة وحيث يُتاح حصول كل المقاطعات.

يحدّد كل ضبط IRQL لمعالج المقاطعات التي يستطيع المعالج إستلامها. وعند إشتغال شعبة في نمط النواة، فإنها ترفع مستوى IRQL لمعالج أو تخفضه. وكما يوضح الشكل (7-7)، تقاطع المعالج المقاطعات من مصدر ذات مستوى IRQL فوق مستوى الضبط الحالي، بينما تمنع أو تحجب المقاطعات، من مصدر ذات مستويات IRQL مساوية للمستوى الحالي أو دونه إلى أن تخفّض شعبة تنفيذ مستوى IRQL.

ترفع شعبة في نمط النواة مستوى IRQL المعالج حيث تشتغل أو تخفضه، وفقاً لما تحاول القيام به. فمثلاً، عند حصول مقاطعة، يرفع مناول المصيدة (أو ربما المعالج) مستوى IRQL لمعالج إلى مستوى IRQL المعيّن لمصدر المقاطعة.

وهذا يمنع المقاطعات عن مستوى IRQL ودونه (على المعالج فقط)، ألا وهو الذي يضمن عدم إنشغال المعالج الذي يخدم المقاطعة بمقاطعة أقلّ أهميّة. وتتمُّ مناولة المقاطعات المحجوبة إما بواسطة معالج آخر أو تنتظر إلى أن يهبط مستوى IRQL. إن عمليّة تغيير مستوى IRQL هي عمليّة قويّة يجب أن تنفّذ بحذر شديد، ولا تستطيع الشعَب في نمط المستعمل تغيير مستوى IRQL المعالج.

مستوى IRQL	نوع المقاطعة
مستوى مرتفع	تدقيق في الماكنة أو خطأ في الناقل العمومي
مستوى الطاقة	إنقطاع الطاقة الكهربائيّة
مستوى مقاطعة المعالج الداخلي	طلب عمل من معالج آخر
مستوى الساعة	ساعة فترات
المستوى n للجهاز	جهاز دخل / خرج بأولويّة أعلى
المستوى 1 للجهاز	جهاز دخل / خرج بأولويّة دنيا
مستوى التوزيع / DPC	توزيع شعبة ومعالجة إستدعاء إجراء مؤجّل (DPC)
مستوى APC	معالجة إستدعاء إجراء لا تزامني (APC)
مستوى منخفض	تنفيذ الشعبة العادي

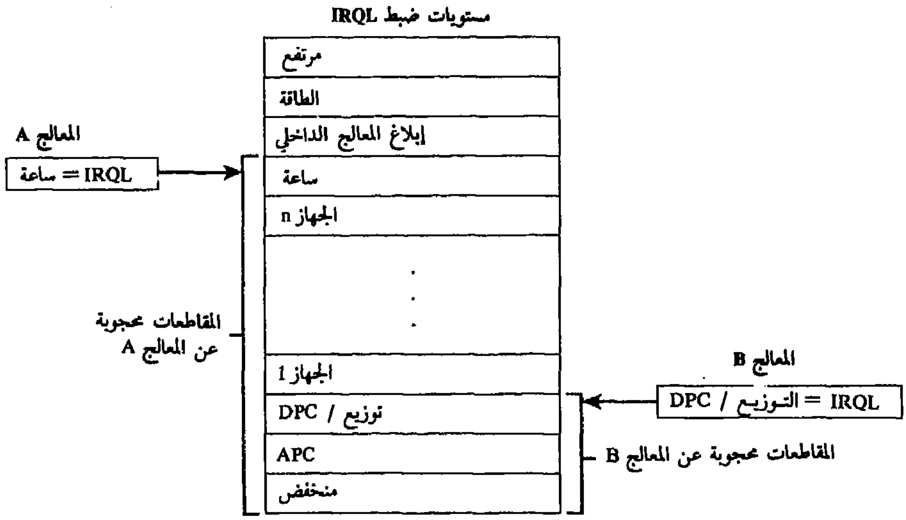
الجدول (7-1)

مستويات طلب المقاطعة (IRQLs)

يتّصف كل مستوى مقاطعة بغرض معيّن. يصدر المعالج مقاطعة مستوى طاقة عندما يكتشف هبوطاً في الفولتيّة من مصدر الطاقة الكهربائيّة. وتؤدّي مقاطعة مستوى طاقة إلى توقُّف نظام التشغيل حيث يسجّل المعلومات الحرّجة المتعلّقة بحالة النظام قبل توقيف نفسه لكي يعاود البدء عند رجوع الطاقة الكهربائيّة والمتابعة من حيث توقّف. (راجع القسم (7-5). لمزيد من المعلومات). تصدر النواة مقاطعة معالج داخلي (IPI) لتطلب من معالج آخر تنفيذ فعل، كتوزيع شعبة معيّنة للتنفيذ أو تحديث مخبأ المخزن المؤقّت لتجاهل الترجمة (TLB) العائد لها. وتصدر ساعة النظام مقاطعة عند فترات منتظمة، وتستجيب النواة عن طريق تحديث الساعة وقياس وقت تنفيذ الشعبة. وإذا كان المعالج يدعم ساعتين، تضيف النواة مستوى مقاطعة ساعة آخر لقياس الأداء. توفّر النواة عدداً من مستويات المقاطعة لتستعمل من قبل الأجهزة المدارة بمقاطعة. ويختلف العدد الفعلي وفقاً لتشكيل المعالج والنظام. وتستعمل النواة مقاطعات البراجيات عند مستويات IRQL للتوزيع / DPC وAPC، لتحفيز جدولة شعبة وللفصل اللاتزامني إلى تنفيذ شعبة. يتمّ وصف مقاطعات البراجيّات في القسم 7-3-2-3.

7-3-2-2 معالجة المقاطعة:

عند حصول مقاطعة، يحفظ مناول المصيدة حالة الماكنة ثم يستدعي موزّع المقاطعة. يرفع فوراً موزّع المقاطعة مستوى IRQL للمعالج إلى مستوى مصدر المقاطعة ليحجب المقاطعات عند ذلك المستوى ودونه خلال خدمة المقاطعة.

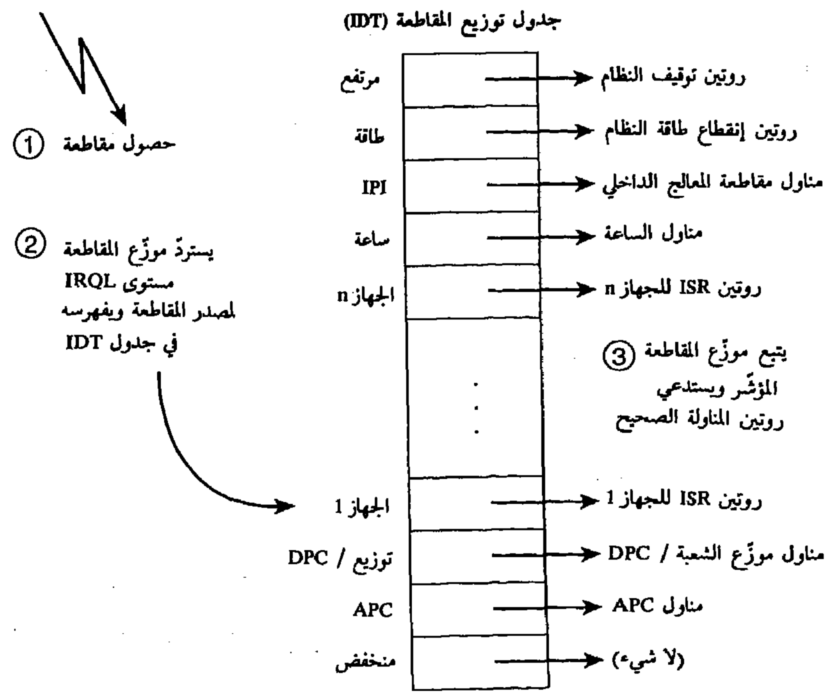


الشكل (7-7)
حجب المقاطعات

وكسائر أنظمة التشغيل المتعدّدة، يستعمل النظام NT جدول توزيع المقاطعة (IDT) لتحديد الروتين لمناولة مقاطعة معيّنة. يخدم مستوى IRQL لمصدر مقاطعة كفهرس جدولي وتشير إدخالات الجدول إلى روتينات مناولة المقاطعة، كما يظهر في الشكل (7-8).

بعد تنفيذ روتين الخدمة، يخفّض موزّع المقاطعة مستوى IRQL للمعالج إلى حيث كان قبل حصول المقاطعة ثم يحمّل حالة الماكنة المحفوظة. وتعاود الشعبة المقاطعة التنفيذ من حيث توقّفت. وعندما تخفّض النواة مستوى IRQL، يمكن أن تخدم المقاطعات بأولويّة أدنى والتي كانت محجوبة. وإذا حصل ذلك، تكرّر النواة المعالجة لمناولة المقاطعة الجديدة.

يحتوي كل معالج على جدول توزيع مقاطعة مستقلّ بحيث تتمكّن المعالجات المختلفة من تشغيل روتينات ISR مختلفة، عند الضرورة. فمثلاً، في نظام معالجة متعدّدة، يستلم كل معالج مقاطعة الساعة لكن معالج واحد فقط يحدّث ساعة النظام إستجابة لهذه المقاطعة. ولكن، تستعمل كل المعالجات المقاطعة لقياس كميّة الوقت ولتحفيز إعادة الجدولة عند إنتهاء كميّة



الشكل (7-8)
خدمة مقاطعة

الوقت المحدّد لشعبة. وبشكل مشابه، قد تتطلّب بعض تشكيلات النظام مناولة مقاطعات جهاز معيّنة من قبل معالج معيّن.

تتواجد معظم الروتينات التي تتناول المقاطعات في النواة. فالنواة، مثلاً، تحدّث وقت الساعة وتوقّف النظام عند حصول مقاطعة مستوى طاقة. لكن العديد من المقاطعات يصدر من قبل أجهزة خارجيّة مثل لوحات المفاتيح وأجهزة التأشير وسوّاقات الأقراص. لذلك، تحتاج مسيقات الأجهزة لوسيلة لإبلاغ النواة عن الروتينات التي يجب إستدعاءها عند حصول مقاطعة جهاز.

توفّر النواة آلية نقّالة ــ وهي كائن تحكُّم بالنواة يسمى كائن مقاطعة ــ يتيح لمسيقات الأجهزة تسجيل روتينات ISR لأجهزتها. يحتوي كائن مقاطعة كل المعلومات التي تحتاجها النواة لربط روتين ISR لجهاز مع مستوى معيّن من المقاطعة، بما في ذلك عنوان الروتين ISR ومستوى IRQL حيث يقاطع الجهاز والإدخال في جدول IDT للنواة حيث يجب ربط روتين ISR.

يسمى ربط روتين ISR مع مستوى معيّن من المقاطعة، توصيل كائن مقاطعة، ويسمى فصل روتين ISR من إدخال جدول IDT، فصل كائن مقاطعة. تتيح هذه الوظائف، التي تحقّق عن طريق إستدعاء وظيفة نواة، لمسيّق جهاز تفعيل روتين ISR عند تحميل المسيّق في النظام وإلغاء تفعيله عند إلغاء تحميل المسيّق.

يمنع إستعمال كائن المقاطعة لتسجيل روتين ISR مسيّقات الأجهزة من العبث مباشرة بعتاد المقاطعة التي تختلف وفقاً لتصاميم المعالج ومن الحاجة لمعرفة أيّة تفاصيل حول جدول توزيع المقاطعة. تساعد مزيّة النواة هذه على إنشاء مسيّقات أجهزة نقّالة لأنها تزيل الحاجة لتشفير لغة assembly أو إظهار إختلافات المعالج في مسيّقات الأجهزة.

توفّر كائنات المقاطعة فوائد أخرى أيضاً. فبإستعمال كائن المقاطعة، تستطيع النواة مزامنة تنفيذ روتين ISR مع الأجهزة الأخرى لمسيّق جهاز يمكن أن يشارك البيانات مع روتين ISR. (راجع الفصل الثامن «نظام الدخل / الخرج» لمزيد من المعلومات). إضافة لذلك، تتيح كائنات المقاطعة للنواة إستدعاء أكثر من روتين ISR واحد بسهولة لأي مستوى مقاطعة. وإذا أنشأت مسيّقات أجهزة متعدّدة كائنات مقاطعة وأوصلتها إلى نفس إدخال جدول IDT، يستدعي موزّع المقاطعة كل روتين عند حصول مقاطعة عند ذلك المستوى. هذا الأمر يتيح للنواة دعم تشكيلات السلسلة المرصّعة بسهولة حيث تقاطع عدّة أجهزة عند نفس الأولويّة.

3-2-3-7 مقاطعات البراجيّات:

رغم أن العتاد تصدر معظم المقاطعات، تصدر النواة NT أيضاً مقاطعات براجيّات لمجموعة متنوّعة من المهام:

- تحفيز توزيع شعبة.
- مناولة نفاذ الوقت.
- التنفيذ اللاالتزامني لإجراء في سياق شعبة معيّنة.
- دعم عمليّات الدخل / الخرج اللاالتزامنيّة.

يتبع وصف لهذه المهمات.

مقاطعات التوزيع: لقد تعرّفت إلى مكان حيث تستعمل النواة مقاطعات البراجيّات ــ في موزّع الشعبة. عند تعذّر مواصلة شعبة التنفيذ، لأنها أنهيت أو لأنها تنتظر مقبض كائن، تستدعي النواة الموزّع مباشرة ليقوم بتبديل سياقي فوراً. لكن وفي بعض الأحيان، تكتشف النواة أنه يجب إعادة الجدولة عندما تكون في عمق طبقات الشيفرة المتعدّدة. في هذه الحالة، يكون الحل الأمثل في طلب التوزيع لكن مع تأجيل حصوله إلى أن تتمّ النواة نشاطها الحالي. واستعمال مقاطعة براجيّات هو الحل الأنسب لهذه المشكلة.

ولأغراض المزامنة (راجع القسم 7-4) ترفع النواة دائماً مستوى IRQL للمعالج إلى مستوى التوزيع / DPC أو فوقه عندما تشتغل. الأمر الذي يحجب مقاطعات البراجيّات (ويلغي تمكين توزيع الشعبة). وعندما تكتشف النواة أنه يجب حصول توزيع، فإنها تطلب مقاطعة مستوى توزيع / DPC لكن وبما أن المستوى IRQL موجود فوق ذلك المستوى، يحتفظ المعالج بالمقاطعة قيد التدقيق. وعندما تتمّ النواة نشاطها الحالي، فإنها تخفّض مستوى IRQL إلى دون مستوى التوزيع / DPC حيث تنشّط مقاطعة التوزيع.

مقاطعات إستدعاء إجراء مؤجّل (DPC): إن تنشيط الموزّع بإستعمال مقاطعة براجيّات هي إحدى الطرق لتأجيل التوزيع إلى أن تصبح الظروف ملائمة. يستعمل النظام NT المقاطعات لتأجيل أنواع أخرى من المعالجات أيضاً.

يتمّ التوزيع عند مستوى IRQL للتوزيع / DPC. وتمرُّ المقاطعات التي تحصل عند هذا المستوى عبر مناول المصيدة إلى الموزّع الذي ينفّذ جدولة الشعَب. وخلال ذلك، تعالج النواة أيضاً روتينات إستدعاء الإجراءات المؤجّلة (DPC). وروتين DPC هي وظيفة تنفّذ مهام نظام أقل أهميّة من المهمّة الحاليّة، تسمى الوظائف «مؤجّلة» لأنه لا يمكن تنفيذها فوراً. وبشكل

مشابه لمقاطعات التوزيع، تنفّذ روتينات DPC فقط بعد أن تنتهي النواة (أو غالباً نظام الدخل / الخرج) من أعمال أكثر أهمّيّة وتخفّض مستوى IRQL إلى دون مستوى التوزيع / DPC.

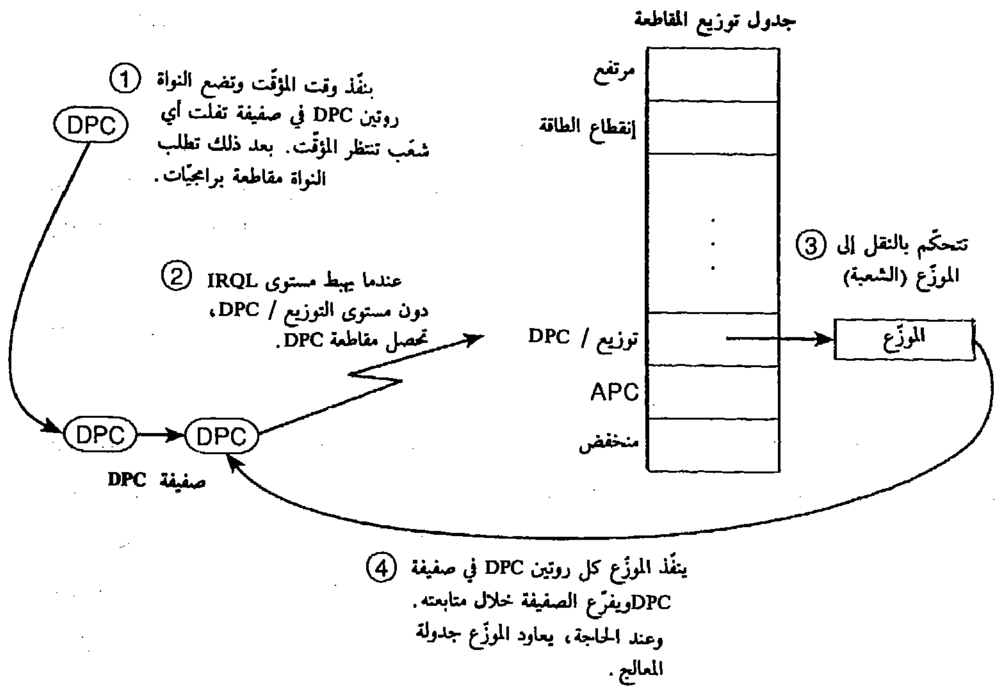
توفّر روتينات DPC لنظام التشغيل القدرة على إصدار مقاطعة وتنفيذ وظيفة نظام في نمط النواة. تستعمل النواة روتينات DPC لمعالجة نفاذ الوقت (وإفلات الشعَب التي تنتظر المؤقّتات) ولإعادة جدولة المعالج بعد نفاذ كميّة وقت الشعبة. تستعمل مسيّقات الأجهزة روتينات DPC لإتمام طلبات الدخل / الخرج. (راجع الفصل الثامن، «نظام الدخل / الخرج» لمزيد من المعلومات).

يمثّل روتين DPC بواسطة كائن DPC وهو كائن يتحكّم بالنواة غير مرئي للبرامج في نمط المستعمل لكنّه مرئي لمسيّقات الأجهزة وشيفرة النظام الأخرى. إن أهمّ قطعة معلومات يحتويها كائن DPC هي عنوان وظيفة النظام التي ستستدعيها النواة عندما تعالج مقاطعة DPC. تخزّن روتينات DPC التي تنتظر التنفيذ في الصفيفة المدارة من قبَل النواة والتي تسمى صفيفة DPC. ولطلب روتين DPC، تستدعي شيفرة النظام النواة لتحفيز كائن DPC ثم وضعه في صفيفة DPC.

يحثّ وضع روتين DPC في صفيفة DPC النواة لطلب مقاطعة براجيّات عند مستوى التوزيع / DPC. ولأن روتينات DPC توضع في صفيفة عادة من قبل البراجيّات التي تشتغل عند مستوى IRQL أعلى، لا تظهر المقاطعة المطلوبة إلى أن تخفّض النواة المستوى IRQL إلى مستوى APC أو مستوى منخفض. يتمّ تصوير معالجة DPC في الشكل (7-9).

تنفّذ روتينات DPC سرياً، أي عندما يهبط مستوى IRQL فإنها تنفّذ دون إعتبار للشعبة الشغّالة. ولأن الشعَب في نمط المستعمل تنفّذ عند مستوى IRQL منخفض، فإنه يحتمل أن يقاطع روتين DPC تنفيذ شعبة مستعمل عادية. وهذا يعني، مثلاً، أنه يمكّن تنفيذ روتين DPC في فسحة العنوان مع التمكّن من الوصول إلى الموارد، دون علمك بذلك. ولهذا ولأنها تنفذ عند مستوى IRQL للتوزيع / DPC، لا تستطيع روتينات DPC الحصول على موارد النظام أو تعديل الذاكرة الظاهريّة للشعبة المستعارة. ويمكنها إستدعاء وظائف النواة، لكن لا يمكنها إستدعاء خدمات النظام وإصدار أخطاء صفحة أو إنشاء كائنات أو إنتظارها. لحسن الحظ، تستطيع فقط شيفرة النظام وضع DPC في صفيفة ويضمن نظام التشغيل تصرُّف روتينات DPC بشكل صحيح. (يجب أن تتأكّد مسيقات الأجهزة من إستعمالها بشكل صحيح أيضاً).

تتوفّر روتينات DPC مبدئياً لمسيّقات الأجهزة، لكن النواة تستعملها أيضاً. فالنواة تستعمل في غالب الأحيان روتين DPC لمناولة نفاذ كميّة الوقت. وعند كل ثانية من ساعة النظام، تحصل



الشكل (7-9)
تسليم روتينات DPC

مقاطعة عند مستوى IRQL للساعة. يحدّث مناول مقاطعة الساعة (الذي يشتغل عند مستوى IRQL للساعة) وقت النظام ثم يخفّض العدّاد الذي يتابع كميّة وقت إشتغال الشعبة الحاليّة. وعندما يصل العدّاد إلى الصفر، تكون كميّة وقت الشعبة قد نفذت وقد تحتاج النواة لإعادة جدولة المعالج، وهي مهمّة بأولويّة أدنى يجب أن تنفّذ عند مستوى IRQL للتوزيع / DPC. يضع مناول مقاطعة الساعة روتين DPC في صفيفة لتحفيز توزيع الشعبة ثم إنهاء عمله وتخفيض مستوى IRQL للمعالج. ولأن مقاطعة DPC تتّصف بأولويّة أدنى من مقاطعات الجهاز، يتمّ مناول مقاطعات الجهاز المعلّقة قبل حصول مقاطعة DPC.

مقاطعات إستدعاء إجراء لاتزامني (APC): عندما تضع النواة كائن DPC في صفيفة، تفصل مقاطعة DPC الناشئة لتنفيذ الشعبة الشغّالة. ومن المفيد أحياناً التمكّن من مقاطعة شعبة معيّنة وتوجيهها لتنفيذ إجراء.

تنفّذ النواة وسائل تنفيذ ذلك بواسطة ما يسمّى إستدعاء إجراء لاتـزامني (APC). وتستطيع كل من شيفرة النظام وشيفرة نمط المستعمل من وضع APC في صفيفة رغم كون

روتينات APC في نمط النواة أكثر قوّة. ومثل DPC، تنفّذ إجراء APC لاتزامنياً عندما تكون الظروف مؤاتية، ولروتينات APC في نمط المستعمل، فالشروط المفروضة هي كالتالي:

- يجب أن تكون الشعبة التي يجب أن تنفّذ روتين APC شغّالة.
- يجب أن يكون مستوى IRQL للمعالج عند مستوى منخفض.
- يجب أن تعلن الشعبة المقصد لروتين APC في نمط المستعمل، أنها مستعدّة (وهو موضوع سيُشرح بعد قليل).

لا تحتاج روتينات APC في نمط النواة إلى «إذن» من الشعبة المقصد لتشتغل في سياق هذه الشعبة، كما تفعل روتينات APC في نمط المستعمل. فروتينات APC في نمط النواة تستطيع مقاطعة شعبة وتنفيذ إجراء دون تدخُّل الشعبة أو موافقتها.

يضع برنامج روتين APC في صفيفة لشعبة معيّنة عن طريق إستدعاء النواة، إما مباشرة (لشيفرة النظام) أو غير مباشرة (لشيفرة نمط المستعمل). وتطلب النواة بدورها مقاطعة برامجيّات عند مستوى APC وعند موافاة كل الشروط المُسرَدة أعلاه، تقاطع الشعبة المقصد وتنفّذ روتين APC.

وكروتينات DPC، يتمّ وصف روتينات APC من قبل كائن تحكّم بنواة يسمى كائن APC. تستقرُّ روتينات APC التي تنتظر التنفيذ في صفيفة APC المدارة من قبَل النواة. وبعكس صفيفة DPC التي هي على إمتداد النظام، فإن صفيفة APC خاصة بالشعبة ــ فكل شعبة تحتوي صفيفة APC خاصة بها. وعند الطلب منها وضع روتين APC في صفيفة، تدرجه النواة في الصفيفة العائدة للشعبة التي ستنفّذ روتين APC.

ولأن APC يُنفّذ في سياق شعبة معيّنة ولأنه ينفّذ عند مستوى IRQL منخفض فإنه لا يعمل بظل نفس تقييدات DPC، ويستطيع الحصول على موارد (كائنات) وإنتظار مقابض كائن وجلب أخطاء صفحة وإستدعاء خدمات النظام. وهذا ما يجعل من روتينات APC مفيدة حتى لشيفرة نمط المستعمل.

ورغم أن شيفرة نمط المستعمل لا تستطيع إنشاء كائن APC أو وضعه في صفيفة مباشرة، تقبل بعض خدمات NT المحليّة روتين APC في نمط المستعمل كبارامتر. فمثلاً، يستطيع نظام فرعي أو DLL تحديد روتين APC عند ضبط المؤقّت. وعندما يتوقّف المؤقّت، تضع النواة روتين APC مجدداً في صفيفة للنظام الفرعي الذي ينفّذه. وإذا وفّر نظام فرعي قدرة APC في النظام NT إلى تطبيقات المستضاف، قد يستعمل تطبيق روتينات APC لتنفيذ تجميع النفايات عند فترات منتظمة. وبشكل مشابه، تتّخذ خدمات الدخل / الخرج المحليّة روتين APC خياري

كبارامتر والذي يتيح لمستدعي تنفيذ روتين وفقاً لنتيجة عملية الدخل / الخرج. (رغم أن النظام الفرعي Win 32 لا يصدر روتينات APC في NT مباشرة إلى روتين API إلا أنه يوفّر قدرات APC في روتينات API () ReadFileEX و () WriteFileEx).

رغم أنها لا تستطيع منع روتينات APC في نمط النواة، تستطيع الشعبة منع تسليم روتينات APC في نمط المستعمل: وفي الواقع، قد تشير شعبة إلى قبولها مقاطعة APC في نمط المستعمل عن طريق إعلان نفسها مستعدّة. ويمكنها القيام بذلك إما بإنتظار مقبض كائن وتحديد أن إنتظارها مستعدّ أو بإختبارها مباشرة لجهة وجود روتين APC معلّق. في كلتا الحالتين، وإذا كان روتين APC في نمط المستعمل معلّقاً، تقاطع النواة (تنبّه) الشعبة، وتنقل التحكّم إلى روتين APC وتعاود تنفيذ الشعبة عند إتمام روتين APC.

يستعمل البرنامج التنفيذي NT روتينات APC في نمط النواة لتنفيذ عمل نظام التشغيل الواجب تنفيذه ضمن فسحة عنوان (في سياق) شعبة معيّنة. ويستطيع إستعمال روتينات APC في نمط النواة للطلب من شعبة توقيف تنفيذ خدمة نظام قابل للمقاطعة، على سبيل المثال، أو لتسجيل نتائج عمليّة دخل / خرج لا تزامنيّة في فسحة عنوان الشعبة. تستعمل الأنظمة الفرعيّة لمحيط روتينات APC في نمط النواة لتعليق شعبة أو لإنهاء نفسها أو للحصول على سياق تنفيذ في نمط المستعمل أو لضبطه. يشرح مجدّداً الفصل الثامن، «نظام الدخل / الخرج» موضوع روتينات APC لأنها مستعملة بكثرة في معالجة الدخل / الخرج في NT.

3-3-7 توزيع الإستثناء:

الإستثناءات بعكس المقاطعات التي تحصل في أي وقت، هي نتيجة مباشرة لتنفيذ برنامج شغّال. تعرّف اللغة C من Microsoft تصميم براجيّات معروف بالمناولة الإستثنائيّة البنيويّة على أنه الذي يتيح للتطبيقات الإستجابة للإستثناءات بتناسق. لقد عرض الفصل الثاني، «نظرة شاملة حول النظام»، المبادىء الأساسيّة للمناولة الإستثنائيّة البنيويّة. يعالج القسم الفرعي هذا الموضوع من وجهة نظر أخرى ـــ حول كيفيّة رؤية النواة لإستثناء وعمّا تفعله عند حصول واحد.

تخدم كل الإستثناءات، ما عدا تلك البسيطة منها التي تحلّ بواسطة مناول المصيدة، من قبل وحدة نواة تسمى موزّع الإستثناء. (راجع الشكل (7-6)). تعتمد وحدة النواة هذه على تصميم المعالج، لكنها تُكتَب باللغة C. أما وظيفة موزّع الإستثناء فهي إيجاد مناول إستثناء يستطيع «التخلّص» من الإستثناء. إن ما يلي هي الإستثناءات التي تعتمد على التصميم والتي تعرّفها النواة:

مخالفة الوصول إلى الذاكرة	قسّمت عدداً كاملاً إلى صفر
فيض عدد كامل	فيض / غيض نقطة عائمة
قسّمت نقطة عائمة على صفر	عامل محتجز لنقطة عائمة
نقطة فصل مزيل العلل	عدم إستقامة من نوع البيانات
تعليمات غير قانونيّة	تعليمات بأفضليّة
خطوة واحدة لمزيل العلل	مخالفة صفحة وقاية
خطأ قراءة صفحة	تجاوز حصّة ملفّ الترتيب في صفحات

تحتجز النواة NT وتتناول بعض هذه الإستثناءات المجهولة لبرامج المستعمل. فمثلاً، تؤدّي مواجهة نقطة فصل مزيل علل خلال تنفيذ برنامج قيد إزالة العلل إلى إصدار إستثناء تتناوله النواة بواسطة إستدعاء مزيل العلل. تتناول النواة إستثناءات أخرى معيّنة عن طريق إرجاع شيفرة حالة غير ناجحة إلى المستدعي.

يُتاح لبعض الإستثناءات العودة دون تعديلها إلى نمط المستعمل. فمثلاً، تصدر مخالفة وصول إلى ذاكرة أو فيض حسابي إستثناء لا يتناوله نظام التشغيل. يستطيع نظام فرعي لمحيط أو تطبيق محلي إنشاء مناولات إستثناء تعتمد على الإطار لمعالجة هذه الإستثناءات عن طُريق إستعمال إيعازات لغة بمستوى مرتفع مصمّمة خصيصاً لمناولة الإستثناءات. إن اللغة C من Microsoft هي أول لغة من Microsoft تدعم المناولة الإستثنائيّة البنيويّة لكن قدرات المناولة الإستثنائيّة في Windows NT ليست خاصة بلغة.

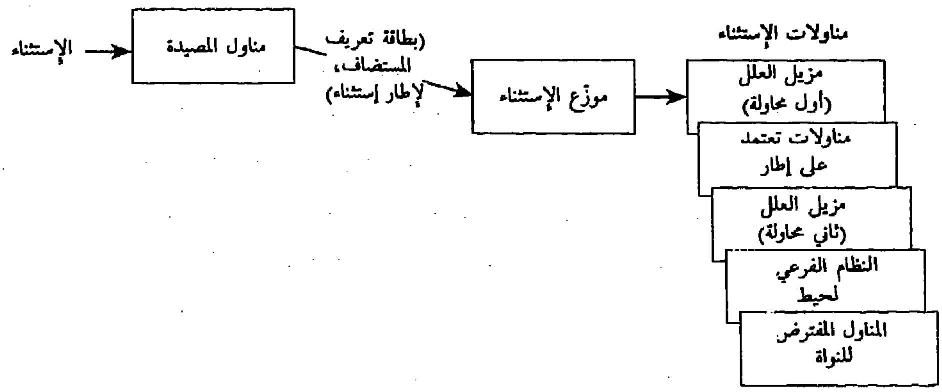
ينسب التعبير «تعتمد على إطار» إلى علاقة مناول الإستثناء مع تنشيط إجراء معينّ. فعند تحفيز إجراء، يوضع إطار تكديس يمثّل تنشيط الإجراء على التكديس. يمكن أن يحتوي إطار تكديس على مناول إستثناء واحد أو أكثر متعلّق به، حيث كل منها يحمي كتلة شيفرة معيّنة في برنامج المصدر. وعند حصول إستثناء، تبحث النواة عن مناول إستثناء متعلّق بإطار التكديس الحالي. وإذا لم يتواجد أي منها، تبحث النواة عن مناول إستثناء متعلّق بإطارالتكديس السابق وهكذا إلى أن تجد مناول إستثناء يعتمد على إطار. وإذا لم يوجد مناول إستثناء، تستدعي النواة مناولات الإستثناءات المفترضة الخاصة بها. (لاحظ أن المناولة الإستثنائيّة تستخدم بشكل مختلف على المعالجات المختلفة. فالمعالج Intel x86 يستعمل طريقة إطار تكديس بينها يستعمل المعالج MIPS R4000 طريقة تعتمد على جدول).

عند حصول إستثناء، صادر عن براجيّات أوعن عتاد، تبدأ سلسلة من الأحداث في النواة. فالتحكُّم ينتقل إلى مناول المصيدة الذي ينشيء إطار مصيدة (كما يفعل عند حصول

مقاطعة). يتيح إطار المصيدة للنظام المعاودة من حيث توقف إذا عولج الإستثناء. كذلك، ينشىء مناول المصيدة سجل إستثناء يحتوي سبب الإستثناء والمعلومات المتعلّقة الأخرى.

إذا حصل الإستثناء في نمط النواة، يُستدعى موزّع الإستثناء روتينياً لتحديد مناول إستثناء يعتمد على إطار لمناولة الإستثناء. ولأن الإستثناءات في نمط النواة غير المعالجة تعتبر خطأً فادحاً في نظام التشغيل، يمكن دائماً توقُّع الموزّع إيجاد مناول إستثناء.

إذا حصل الإستثناء في نمط المستعمل، يقوم موزّع الإستثناء بأمر أكثر إتقاناً. وكما تذكر من الفصل الرابع، «المعالجات والشعَب»، فكان النظام الفرعي لمحيط يستطيع إنشاء منفذ مزيل علل ومنفذ إستثناء لكل معالجة ينشئها. تستعمل النواة هذه المنافذ في مناولتها الإستثنائيّة المفترضة، كما يوضح ذلك الشكل (7-10).



الشكل (7-10)
توزيع إستثناء

إن نقاط فصل مزيل العلل هي المصادر العامة للإستثناءات. لذلك، أول عمل يقوم به موزّع الإستثناء هو إرسال رسالة (عبر LPC) إلى منفذ مزيل العلل المتعلّق بالمعالجة التي تجلب الإستثناء. هذا الأمر يتيح للمستعمل مناولة بنيات البيانات وإصدار أوامر مزيل العلل.

إذا لم يسجل أي منفذ لمزيل العلل أو إذا لم يتناول مزيل العلل الإستثناء، يبدل موزّع الإستثناء النمط إلى نمط المستعمل ويستدعي روتينياً لإيجاد مناول إستثناء يعتمد على إطار. وإذا لم يوجد أي واحد أو إذا لم يتناول أي منها الإستثناء، يبدّل موزّع الإستثناء النمط مجدّداً إلى نمط النواة ويستدعي مزيل العلل مجدداً ليتيح للمستعمل تنفيذ المزيد من عمليات إزالة العلل.

إذا مزيل العلل لا يشتغل ولا يوجد أي مناول إستثناء يعتمد على إطار، ترسل النواة رسالة إلى منفذ الإستثناء المتعلّق بمعالجة الشعبة. وقد تمّ تسجيل منفذ الإستثناء هذا، في حال وجد، من قبل النظام الفرعي للمحيط الذي يتحكّم بهذه الشعبة. يوفّر منفّذ الإستثناء للنظام الفرعي للمحيط، الذي يفترض أن يكون مستمعاً عند المنفذ، إمكانيّة ترجمة الإستثناء NT إلى إشارة أو إستثناء خاص بمحيط. فمثلاً، وعندما يصل إلى POSIX رسالة من النواة تشير إلى إصدار إستثناء من قبل إحدى شعبه، يرسل النظام الفرعي POSIX إشارة على شكل POSIX إلى الشعبة التي أصدرت الإستثناء.

ورغم أن النظام الفرعي POSIX يربط إفتراضياً منفّذ إستثناء مع كل من معالجاته، قد لا تزوّد الأنظمة الفرعيّة الأخرى منفذاً أو قد لا تقوم بأي عمل عندما تبلغها النواة عن وجود إستثناء غير مناول في معالجة الإستثناء ولم يتناول النظام الفرعي الإستثناء، تنفّذ النواة مناول إستثناء مفترض ينهي المعالجة ذات الشعبة التي أدّت إلى الإستثناء.

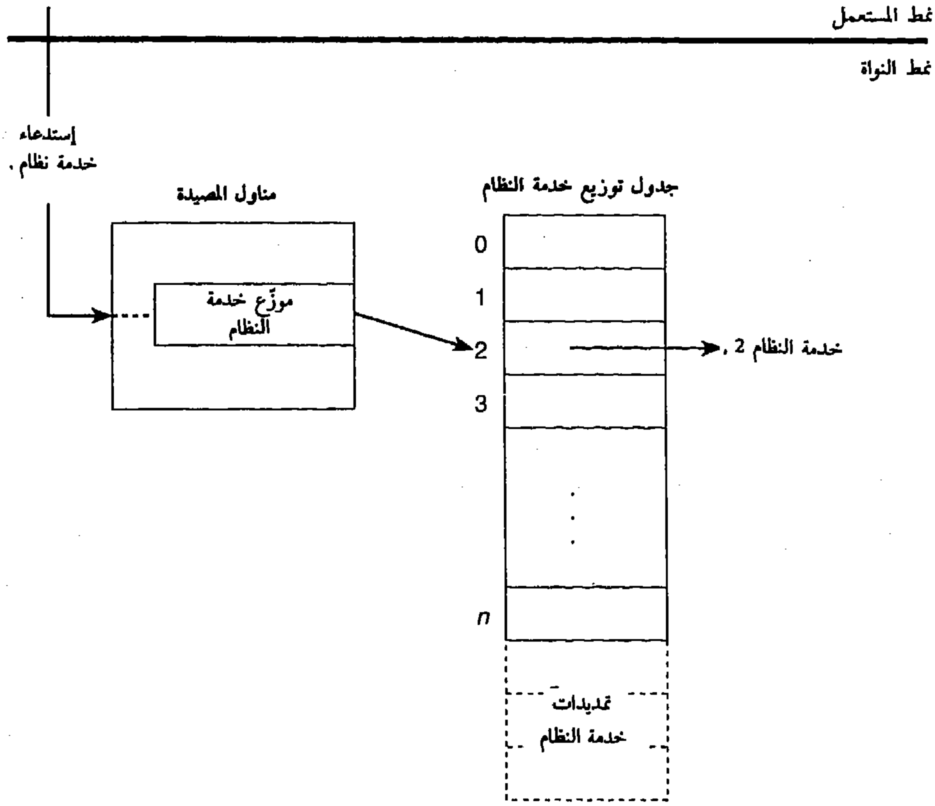
4-3-7 توزيع خدمة النظام:

كما أوضح الشكل (7-6)، يوزّع مناول مصيدة النواة NT المقاطعـات، والإستثناءات وإستدعاءات خدمة النظام. ولقد شرحت الأقسام السابقة مناولة المقاطعة والإستثناء، وهذا القسم يعالج بإختصار خدمات النظام. إن إستدعاءات خدمة النظام، التي تصدر المصايد المعالجة كإستثناءات في النظام Windows NT، جديرة بالذكر لناحية مدوديّة النظام. وتتيح الطريقة التي تستخدم فيها النواة خدمات النظام إضافة خدمات جديدة دينامية إلى نظام التشغيل في الإصدارات المستقبليّة.

فعندما تستدعي شعبة في نمط المستعمل خدمة نظام، يتاح فجأة للشعبة تشغيل شيفرة نظام التشغيل ذات الأفضليّة. وقد تعبث شعبة في نمط المستعمل ببنيات بيانات النظام أو تنقل أشياءً في الذاكرة، وتشوّش على النظام أو على المستعملين الآخرين. لهذا السبب، توفّر المعالجات عادة تعليمات خاصة فقط لخدمات النظام. تصدر التعليمات ـ syscall على معالجات MIPS و int 2Eh على معالجات Intel x86 ـ عندما تستدعي شعبة في نمط المستعمل خدمة نظام. وتصدر العتاد مصيدة وتبدّل التنفيذ من نمط المستعمل إلى نمط النواة. وعند حصول ذلك، تنسخ النواة المتغيّرات المستقلّة للمستدعي من التكديس في نمط المستعمل للنواة إلى التكديس في نمط النواة العائد لها (لكي لا يتمكّن المستعمل من تغيير المتغيّرات المستقلّة على هواه) ثم تنفذ خدمة النظام.

كما يوضح الشكل (7-11)، تستعمل النواة جدول توزيع لخدمة نظام لإيجاد خدمات

النظام. وهذا الجدول يشبه جدول توزيع المقاطعة الذي شرح سابقاً بإستثناء إحتواء كل إدخال على مؤشّر إلى خدمة نظام عوَضاً عن روتين مناولة مقاطعة.

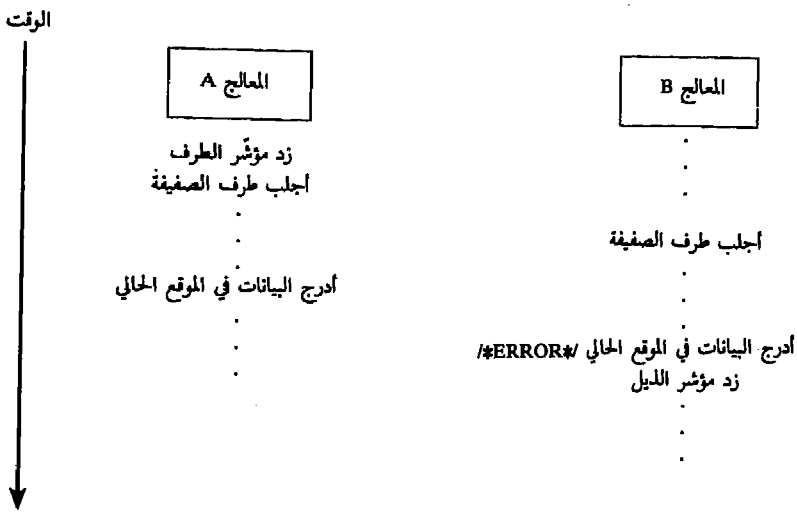


الشكل (11-7)
إستثناءات خدمة النظام

يوفّر إستعمال جدول توزيع خدمة النظام إمكانيّة تمديد خدمات النظام NT المحليّة. وتستطيع النواة دعم خدمات نظام جديد عن طريق توسيع الجدول دون الحاجة للقيام بتغييرات على النظام أو على التطبيقات. بعد كتابة الشيفرة لخدمة جديدة، يستطيع مدير النظام تشغيل برنامج خدماتي ينشىء ديناميا جدول توزيع جديد. وقد يحتوي الجدول الجديد إدخالاً آخر يشير إلى خدمة النظام الجديدة. لكن ورغم عدم توفّر هذه القدرة والتداخل مع المستعمل في الإصدار الأول للنظام Windows NT، إلا أنه يمكن إضافتها لاحقاً.

4-7 مزامنة المعالجات المتعدّدة:

إن مبدأ المنع المتبادل هو مبدأ حرج في تطوير أنظمة التشغيل. وهو ينسب إلى ضمانة شعبة واحدة، وفقط واحدة، من الوصول إلى مورد معينّ في كل مرة. والمنع المتبادل ضروري عندما لا يعير مورد نفسه إلى وصول مشترك أوعندما تؤدي المشاركة إلى نهاية غير متوقّعة. فمثلاً، إذا نسخت إلى شعبتين ملفاً إلى منفذ طابعة في نفس الوقت، قد يكون خرجها مبعثراً. وبشكل مشابه، إذا قرأت شعبة واحدة موقع ذاكرة خلال كتابة أخرى عليه، فإن الشعبة الأولى ستتلقى بيانات غير متوقّعة. بشكل عام، لا يمكن مشاركة الموارد «المكتوبة» دون تقييدات، بينما يمكن مشاركة الموارد غير المعرّضة للتعديل. يوضح الشكل (7-12) ماذا يحصل عندما تكتب شعبتين تشتغلان على معالجين مختلفين إلى صفيفة دائريّة.



الشكل (7-12)
مشاركة غير صحيحة للذاكرة

لأن الشعبة الثانية حصلت على قيمة مؤشّر طرف الصفيفة قبل أن تحدّثه الشعبة الأولى، أدرجت الشعبة الثانية بياناتها في نفس الموقع الذي إستعملته الشعبة الأولى، حيث كتبت فوق البيانات وتركت موقع صفيفة فارغاً. لاحظ أنه رغم أن الشكل يوضح ما قد يحصل على نظام متعدّد المعالجات، يمكن لنفس الخطأ أن يحصل على نظام أحادي المعالج إذا نفذ نظام التشغيل تبديل سياقي إلى الشعبة الثانية قبل أن تحدّث الشعبة الأولى مؤشّر طرف الصفيفة.

تسمى أقسام الصفيفة التي تتمكّن من الوصول إلى مورد غير مشارك، الأقسام الحرجة. ولضمان شيفرة صحيحة، تستطيع شعبة واحدة في كل مرّة التنفيذ في قسم حرج. وخلال كتابة شعبة واحدة إلى ملف أو تحديث قاعدة بيانات أو تعديل متغيّر مشارك، لا يمكن لأية شعبة أخرى الوصول إلى نفس المورد. إن الشيفرة المبيّنة في الشكل (7-12) هي القسم الحرج الذي يتمكّن من الوصول غير الصحيح إلى بنية بيانات مشاركة دون منع متبادل.

المنع المتبادل، رغم أهميّته لكل أنظمة التشغيل، مهمّ بشكل خاص لنظام تشغيل المعالجة المتعدّدة المتناظرة المقرونة بأحكام (SMP) كنظام Windows NT حيث تشتغل نفس شيفرة النظام في نفس الوقت على أكثر من معالج واحد، حيث تشارك بنيات بيانات معيّنة مخزّنة في الذاكرة العامة. وفي النظام Windows NT، تكون وظيفة النواة توفير آليات تستطيع شيفرة النظام إستعمالها لمنع شعبتين من تعديل نفس البنية في نفس الوقت. توفّر النواة المبادىء الأساسيّة للمنع المتبادل التي تستعملها، وبقيّة البرنامج التنفيذي لمزامنة وصولها إلى بنيات البيانات العامة.

يشرح القسم الفرعي التالي كيفيّة إستعمال النواة للمنع المتبادل لحماية بنيات البيانات العامة. يركّز القسم التالي على المنع المتبادل وآليات المزامنة التي توفّرها النواة للبرنامج التنفيذي والتي توفّره بدورها إلى نمط المستعمل.

7-4-1 مزامنة النواة:

في خلال المراحل المتنوّعة لتنفيذها، يجب أن تضمن النواة تنفيذ معالج واحد، وواحد فقط، في كل مرة ضمن القسم الحرج. إن الأقسام الحرجة للنواة هي أقسام الشيفرة التي تعدّل بنية بيانات عامة مثل قاعدة بيانات موزّع النواة أو صفيفة DPC. ولا يستطيع نظام التشغيل العمل بشكل صحيح ما لم تضمن النواة وصول الشعّب إلى بنيات البيانات هذه بطريقة منعيّة متبادلة.

لكن أكبر الإهتمام ينصبُّ على المقاطعات. فمثلاً، قد تحدّث النواة بنية بيانات عامة عند حصول مقاطعة ذات روتين مناولة مقاطعة يعدّل أيضاً البنية. تمنع أنظمة التشغيل الأحاديّة المعالج في بعض الأحيان حصول مثل ذلك عن طريق إلغاء تمكين كل المقاطعات في كل مرة تتمكّن من الوصول إلى البيانات العامة، لكن النواة NT تحتوي على حل أكثر تعقيداً. لكن قبل إستعمال مورد عام، تحجب النواة مؤقتاً هذه المقاطعات ذات مناولات المقاطعة التي تستعمل أيضاً المورد. وهي تفعل ذلك عن طريق رفع مستوى IRQL للمعالج إلى أعلى مستوى مستعمل من قبل أي مصدر مقاطعة جهدي يمكنه الوصول إلى البيانات العامة. فمثلاً، المقاطعة عند مستوى التوزيع / DPC، تؤدّي إلى تشغيل الموزّع الذي يستعمل قاعدة بيانات الموزّع. لذلك،

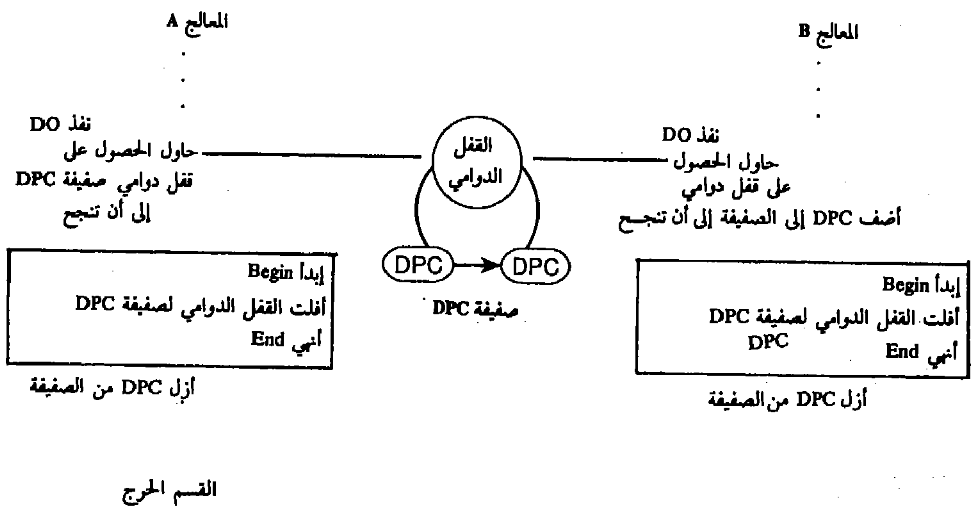
فإن أي جزء آخر من النواة يستعمل قاعدة بيانات الموزّع برفع المستوى IRQL إلى مستوى التوزيع / DPC، حيث يحجب مقاطعات مستوى التوزيع / DPC قبل إستعمال قاعدة بيانات الموزّع.

هذه الإستراتيجيّة تناسب النظام الأحادي المعالج بشكل جيّد، لكنها غير مناسبة لتشكيل متعدّد المعالجات. فرفع مستوى IRQL على معالج واحد لا يمنع حصول مقاطعة على معالج آخر. كذلك تحتاج النواة لضمانة للمنع المتبادل للوصول عبر المعالجات المتعدّدة.

تسمى الآلية التي تستعملها النواة لتحقيق المنع المتبادل للمعالجات المتعدّدة، قفل دوامي. القفل الدوامي هو آلية قفل متعلّقة ببنية البيانات العامة كصفيفة DPC المبيّنة في الشكل (7-13).

وقبل إدخال أي من الأقسام الحرجة المبيّنة في الشكل، يجب أن تحصل النواة على القفل الدوامي المتعلّق بصفيفة DPC المحميّة، فإذا لم يكن القفل الدوامي متوفّراً، تستمرّ النواة في محاولة الحصول على القفل إلى أن تنجح في ذلك. ويسمى القفل الدوامي بهذه التسمية لأن النواة (وبالتالي المعالج) تبقى في دوّامة إلى أن تحصل على القفل.

تستقرّ الأقفال الدوّاميّة، كبنيات البيانات التي تحميها، في الذاكرة العامة. وتكتب شيفرة الحصول على القفل الدوامي وشيفرة إفلاته بلغة assembly للحصول على السرعة واستخدام آلية القفل التي يوفّرها بنية المعالج. وتستخدم الأقفال الدوامية، على العديد من التصاميم، مع



الشكل (7-13)
إستعمال قفل دوامي

عمليات الإختبار والضبط المدعومة من العتاد، والتي تختبر قيمة متغيّر قفل وتحصل على القفل في تعليمة ذريّة واحدة. يمنع إختبار القفل والحصول عليه في تعليمة واحدة شعبة ثانية من الحصول على القفل خلال فترة إختبار الشعبة الأولى للمتغيّر وفترة حصولها عليه.

فعندما تحاول شعبة الحصول على قفل دوامي، تتوقّف كل النشاطات الأخرى على ذلك المعالج. لذلك، لا تشفع أبداً شعبة تحتوي على قفل دوامي لكن يتاح لها مواصلة التنفيذ لكي تفلت القفل بسرعة. تستعمل النواة الأقفال الدواميّة بعناية كبيرة، حيث تخفّض عدد التعليمات التي تنفّذها خلال إحتفاظها بالقفل الدوامي.

توفّر النواة الأقفال الدواميّة للأجزاء الأخرى من البرنامج التنفيذي عبر مجموعة من وظائف النواة. فمثلاً، تحصل مسيقات الجهاز على الأقفال الدواميّة لضمان الوصول إلى مسجلات الجهاز وبنيات البيانات العامة الأخرى من قبل جزء واحد فقط من مسيّق جهاز (ومن معالج واحد فقط) في كل مرة. يشرح هذا الموضوع مرة أخرى في الفصل الثامن، «نظام الدخل / الخرج».

2-4-7 مزامنة البرنامج التنفيذي:

تحتاج أيضاً البراجيّات التنفيذيّة خارج النواة لمزامنة الوصول إلى بنيات البيانات العامة في محيط متعدّد المعالجات. فمثلاً، يحتوي برنامج إدارة VM على قاعدة بيانات إطار صفحة واحدة، التي يتمّ الوصول إليها كبنية بيانات عامة، ويجب أن تضمن مسيّقات الجهاز الوصول إلى البرنامج التنفيذي لأجهزتها. يستطيع البرنامج التنفيذي، عن طريق إستدعاء وظائف النواة، إنشاء قفل دوامي والحصول عليه ثم إفلاته.

لكن الأقفال الدواميّة تلبّي حاجات البرنامج التنفيذي لآليات المزامنة جزئياً فقط. لأن إنتظار قفل دوامي يؤخّر المعالج والأقفال الدواميّة ممكن إستعمالها في الحالات المحدّدة المشدّدة التالية:

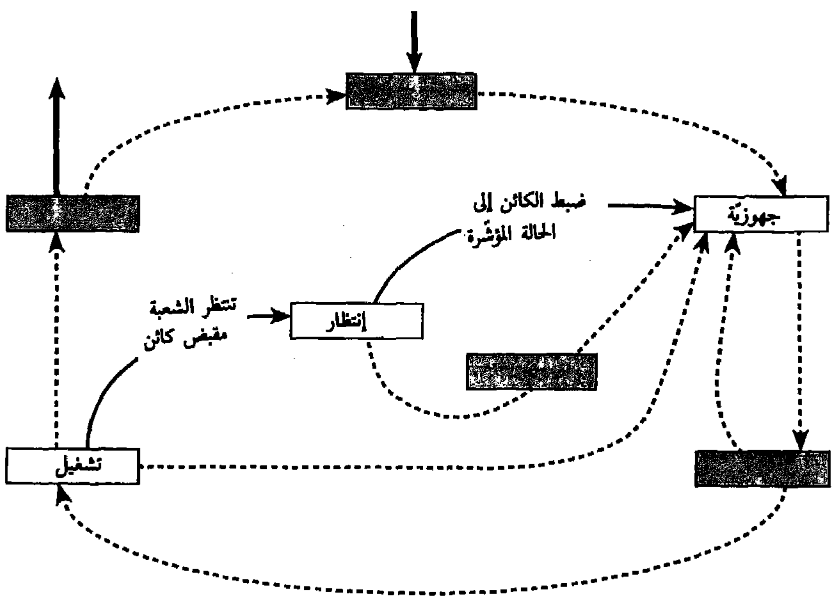
- يجب أن يتمّ الوصول إلى الموارد المحميّة بسرعة دون تفاعلات معقّدة مع الشيفرة الأخرى.
- لا يمكن إخراج شيفرة القسم الحرج من الذاكرة ولا يمكن إنشاء مراجع إلى البيانات المرتّبة في صفحات ولا يمكن إستدعاء إجراءات خارجيّة (بما فيها خدمات النظام) ولا يمكن إصدار مقاطعات أو إستثناءات.

هذه التقييدات محدّدة ولا يمكن تلبيتها بظلّ كل الحالات. إضافة لذلك، يجب أن ينفّذ البرنامج التنفيذي أنواعاً أخرى من المزامنة إضافة إلى المنع المتبادل ويجب أن يوفّر آليات مزامنة لنمط المستعمل.

تزوّد النواة آليات مزامنة إضافيّة إلى البرنامج التنفيذي على شكل كائنات نواة، تعرف جماعياً بإسم كائنات الموزّع. تستطيع شعبة المزامنة مع كائن موزّع عن طريق إنتظار مقبض الكائن. ونتيجة لذلك تعلّق النواة الشعبة وتغيّر حالة الموزّع من التشغيل إلى الإنتظار، كما يوضح ذلك في الشكل (7-14). وتزيل النواة الشعبة من صفيفة جهوزيّة الموزّع ولا تعتبرها قيد التنفيذ.

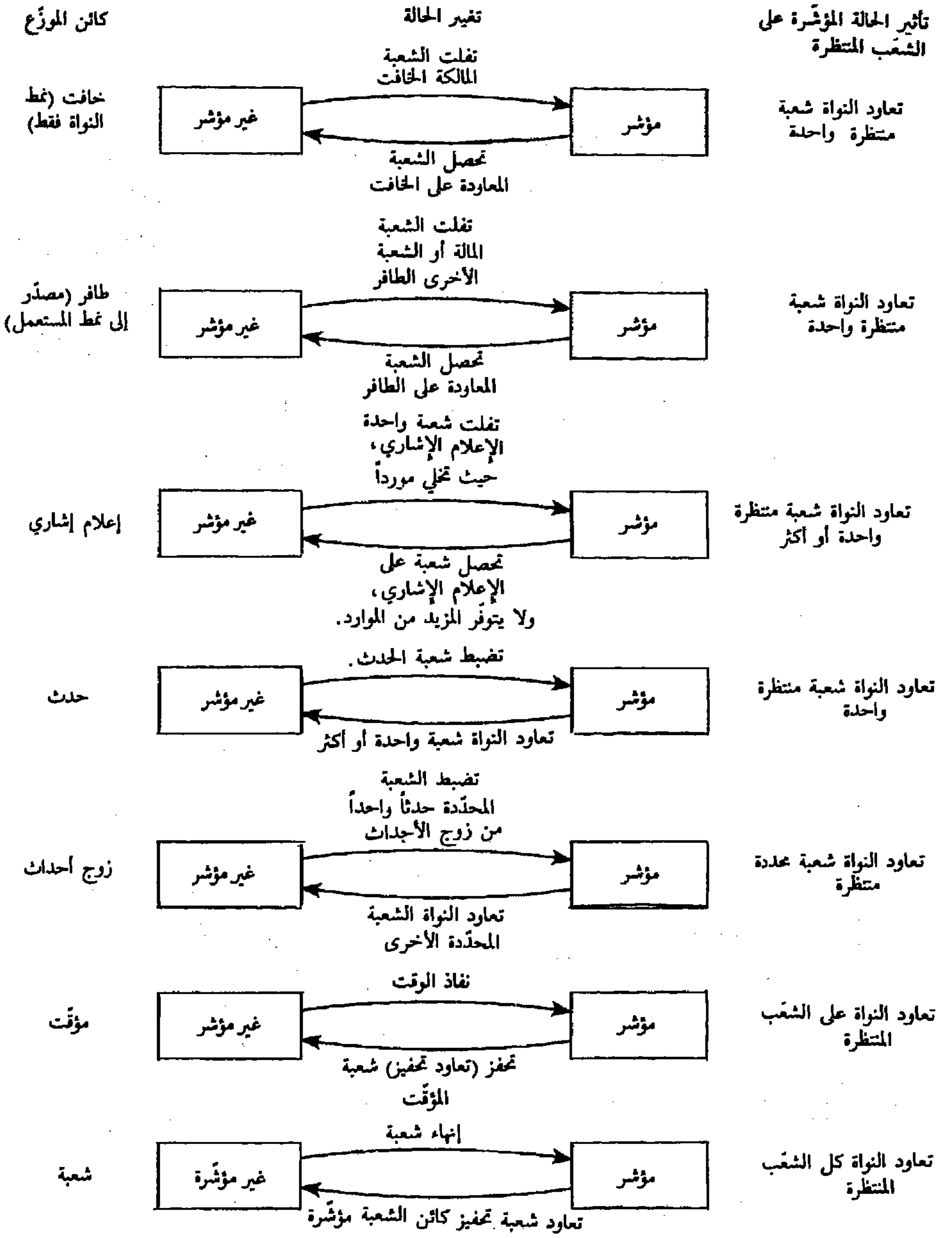
لا تستطيع شعبة معاودة تنفيذها إلى أن تغيّر النواة حالة الموزّع من الإنتظار إلى الجهوزيّة. يحصل هذا التغيير عندما تتغيّر حالة كائن الموزّع التي تنتظر الشعبة مقبضه، من الحالة غير المؤشّرة إلى الحالة المؤشّرة (مثلاً، عندما تضبط شعبة كائن حدثاً). وتكون النواة مسؤولة عن نوعي الإنتقال. توضح بعض كائنات موزّع النواة وأحداث النظام التي تحفز تغيير حالتها في الشكل (7-15) على الصفحة التالية.

يوفّر كل نوع من كائنات الموزّع نوعاً معيّناً من قدرة المزامنة. فمثلاً، توفّر كائنات الخوافت المنع المتبادل، بينما يعمل الإعلام الإستشاري كبوّابة يستطيع عبرها عدد متغيّر من الشعَب المرور ــ وهي مفيدة عند توفّر عدد من الموارد المتطابقة. يمكن إستعمال الأحداث



الشكل (7-14)
إنتظار كائن موزّع

إما لإعلان حصول فعل ما أو لإستخدام المنع المتبادل. إن أزواج الأحداث هي وسائل النواة لدعم روتينات LPC السريعة. وهو الشكل الأمثل لتمرير الرسائل المستعمَل من قبل النظام



الشكل (15-7)

كائنات موزّع النواة المنتقاة

الفرعي Win 32. تتوقّف المؤقّتات عندما تنفذ كميّة وقت الساعة المضبطة. وتستطيع الشعبة إنتظار إنهاء شعبة أخرى، وهو أمر مفيد لمزامنة نشاطات شعبتين متعاونتين. وسوية، توفّر كائنات موزّع النواة للبرنامج التنفيذي مرونة كبيرة في مزامنة تنفيذه.

تحصل كائنات المزامنة المرئيّة للمستعمل التي سبق وشرحت في الفصل الرابع، «المعالجات والشعَب»، على قدرات المزامنة من كائنات موزّع النواة. ويغلّف كل كائن مرئي للمستعمل يدعم المزامنة كائن موزّع نواة واحد على الأقلّ. توضع أمثلة الضبط التالية لحدث كيفيّة تفاعل المزامنة مع توزيع الشعبة:

1 ــ تنتظر شعبة في نمط المستعمل مقبض كائن حدث.

2 ــ تغيّر النواة الحالة المُجَدْوِلَة للشعبة من الجهوزيّة إلى الإنتظار ثم تضيف الشعبة إلى لائحة الشعَب المنتظرة للحدث.

3 ــ تضبط شعبة أخرى الحدث.

4 ــ تستعرض النواة لائحة الشعَب المنتظرة للحدث. فإذا كانت شروط إنتظار شعبة مؤاتية، تغيّر النواة حالة الشعبة من الإنتظار إلى الجهوزيّة. وإذا كانت الشعبة شعبة بأولويّة متغيّرة، قد تعزّز النواة أولويّة تنفيذها.

5 ــ بسبب جهوزيّة شعبة جديدة للتنفيذ، يقوم الموزّع بإعادة الجدولة. وإذا وُجد شعبة شغّالة بأولويّة أدنى من أولويّة الشعبة الجاهزة حديثاً، فإنه يشفع الشعبة بأولويّة أدنى، حيث يصوّر مقاطعة براجميّات لتحفيز تبديل سياقي إلى الشعبة بأولويّة أعلى.

6 ــ إذا لم يكن ممكناً شفع أي معالج، يضع الموزّع الشعبة الجاهزة في صفيفة جهوزيّة الموزّع لجدولتها لاحقاً.

5-7 إستعادة الطاقة الكهربائيّة:

لقد كانت إحدى الأهداف الرئيسيّة في تصميم النظام Windows NT جعله نظام تشغيل إعتمادي قوياً. وقد يسأل المرء إلى أي مدى هذه الإعتماديّة؟ رغم أن تصميم المناولة الإستثنائيّة يساعد على حماية إعتماديّة النظام من الداخل، لكن ماذا يحصل إذا تدخّلت عوامل خارجيّة وهدّدت تكامل نظام التشغيل؟ إحدى هذه المخاطر الخارجيّة هي محاولة المستعملين غير الشرعيين تجاوز الإجراءات الأمنيّة للنظام.

الخطر الآخر هو البرامج التي تشتغل بتهوُّر وتستهلك موارد النظام. يعالج تصميم الأمان في النظام Windows NT الإهتمام الأول وتساعد حدود حصص الموارد على معالجة المشكلة الثانية. لكن، يبقى هناك خطر محدق آخر وهو: إنقطاع الطاقة الكهربائيّة.

تحجز النواة NT ثاني أعلى أولويّة مقاطعة لإنقطاع الطاقة الكهربائيّة. تبلغ مقاطعة إنقطاع الطاقة الكهربائيّة نظام التشغيل عند حصول تغيير في إمداد الطاقة الكهربائيّة لكي يوقف النظام بأفضل طريقة ممكنة. وإذا لم يوقف نظام التشغيل النظام، تفقد كل المعالجات قيد المعالجة عند إنقطاع الطاقة. وقد يتمّ تنفيذ البرامج أو لا يتمّ، وقد يترك الموارد التي إستعملها مثل الملفّات في حالة قابلة للإستعادة أو قد لا يتركها في مثل هذه الحالة.

عند إنقطاع الطاقة الكهربائيّة، يتوفّر لنظام التشغيل ما يكفي من الوقت لتحفيز توقّف منظّم للنظام. وإذا زُوّد الحاسوب ببطاريّة مساندة للذاكرة، يمكن إستعادة البيانات من الذاكرة عند رجوع الطاقة الكهربائيّة. ويمكن إعادة بدء الأعمال التي كانت قيد المعالجة أو مواصلتها وفقاً لحالتها عند إنقطاع الطاقة الكهربائيّة. (وطبعاً، إذا لا تتواجد بطاريّة مساندة، لا يمكن تحقيق هذا النوع من الإستعادة).

لا يكفي إعادة تحميل المسجلات المتطايرة ومعاودة التنفيذ لإستعادة النظام بالكامل. ولأن أجهزة الدخل / الخرج تعمل بإستقلاليّة عن بقيّة نظام التشغيل، فإنها تتطلّب دعم النواة التالي للإستعادة من إنقطاع الطاقة:

- يجب إعادة تحفيزها بعد رجوع الطاقة المقطوعة.
- يجب أن تتمكّن من تحديد حصول إنقطاع في الطاقة الكهربائيّة.

يوفّر كائنا تحكُّم بالنواة هذه القدرات. ويتيح كائن إبلاغ الطاقة لمسيقات الجهاز تسجيل روتين إستعادة الطاقة الكهربائيّة الذي تستدعيه النواة عند رجوع الطاقة الكهربائيّة. يقرّر المسيق ما يجب أن يقوم به الروتين، لكن بشكل عام، فإنّه ينفّذ العمليات مثل إعادة تحفيز جهاز وإعادة بدء عمليات الدخل / الخرج التي تمّ مقاطعتها.

لتسجيل روتين إستعادة الطاقة، ينشىء مسيق كائن إبلاغ الطاقة، حيث يستدعي النواة مجدّداً لإدراج الكائن صفيفة مدارة بنواة. وعند رجوع الطاقة الكهربائيّة، تستعرض النواة الصفيفة، حيث تستدعي كل روتين إستعادة طاقة بترتيب.

تزوّد النواة كائن تحكّم آخر يستعمل من قبل مسيقات الجهاز، يسمى كائن حالة الطاقة. فعن طريق إنشاء كائن حالة الطاقة وإدراجه في صفيفة معرّفة من قبل نواة أخرى، يستطيع

مسيّق جهاز تحديد، قبل تنفيذ العمليات غير المقاطعة (كتخزين البيانات في مسجّل جهاز)، وحصول إنقطاع في الطاقة الكهربائيّة. فإذا حصل ذلك، لا يواصل المسيّق العملية. يوفّر الفصل الثامن معلومات إضافيّة حول هذه المواضيع المتعلّقة بالدخل / الخرج.

6-7 بإختصار:

النواة NT هي محور كل النشاطات في النظام Windows NT. وهي تتحكّم بالمعالج عن طريق جدولة الشعَب للتنفيذ وتوزيعها، والإستجابة للمقاطعات والإستثناءات، واستخدام آليات المزامنة بمستوى منخفض لتستعمل من قبلها ومن قبل الأجزاء الأخرى للبرنامج التنفيذي. تعتمد بقيّة البرنامج التنفيذي NT على الوظائف المتوفّرة من قبل النواة والمبادىء الأساسيّة التي تبني على أساسها سياسات نظام التشغيل وتجعل القدرات متوفّرة لنمط المستعمل.

تتضمّن المبادىء الأساسيّة للنواة سلسلة من الكائنات التي تحتويها كائنات البرنامج التنفيذي. تمكّن كائنات التحكُّم بالنواة مجموعة متنوّعة من وظائف نظام التشغيل الخاصة بينما كائنات موزّع النواة هي مبادىء أساسيّة ذات قدرات مزامنة مركّبة بالداخل. إن المزامنة، داخل النواة وخارجها، وهي حرجة بالنسبة للتشغيل الصحيح لنظام التشغيل. وتكون المهمّة صعبة عندما يشتغل نظام التشغيل على حواسيب متعدّدة المعالج. تزامن النواة تنفيذها لتعمل بشكل صحيح وتتيح الآليات التي توفّرها لبقيّة البرنامج التنفيذي القيام بنفس الشيء.

ومن ضمن مهامها الأخرى، تتيح النواة مساعدة خاصة لنظام الدخل / الخرج. وهي توفّر كائنات ووظائف تستعملها مسيقات الجهاز لمزامنة تنفيذها عبر المعالجات المتعدّدة ولإستعادة عمليات الدخل / الخرج بعد حصول إنقطاع في الطاقة الكهربائيّة. إن نظام الدخل / الخرج وتوصيلاته إلى النواة NT هي مواضيع الفصل التالي.

8

نظام الدخل / الخرج

كتب A. M. Lister في كتابه، «المبادىء الأساسيّة لأنظمة التشغيل»، ما يلي: «إصطلاحياً، يعتبر الدخل / الخرج إحدى المناطق الأكثر شحاحة في تصميم نظام التشغيل حيث أنه مجال يصعب فيه التعميم ويسوّده الطرق الخاصة». وبالفعل، فالصعوبة تكمن في العدد الكبير لأجهزة الدخل / الخرج وطبيعتها المختلفة التي يجب أن يدعمها نظام التشغيل. ويكمن التحدّي الذي يواجهه مصمّم نظام الدخل / الخرج وهو في إنشاء تداخل ظاهري مع أجهزة الدخل / الخرج التي تتيح للمبرمجين إسترداد البيانات أو تخزينها دون مراعاة خصوصيّات الأجهزة الإفراديّة.

يجب على نظام دخل / خرج الذي يستطيع تكثيف الصفيفة الكبيرة من الأجهزة إلى نموذج واحد أن يكون شاملاً. ويجب عليه إستيعاب حاجات الأجهزة الموجودة من الماوس إلى لوحات المفاتيح والطابعات والوحدات الطرفيّة لعرض الرسوم التخطيطيّة ومسيّقات الأجهزة ومسيّقات CD-ROM وحتى الشبكات. ويجب أن يأخذ بعين الإعتبار تقنيّة التخزين والدخل المستقبلين أيضاً. يحمي نظام الدخل / الخرج NT الذي يوفّر تداخلاً متناسقاً بمستوى مرتفع لعمليات الدخل / الخرج بمستوى البرنامج التنفيذي، البرامج التطبيقيّة من أوجه الإختلاف بين الأجهزة الفعليّة. وهو أيضاً يحمي بقيّة نظام التشغيل من تفاصيل مناولة الجهاز، وبالتالي يخفّض الشيفرة المعتمدة على العتاد ويعزلها.

صمّم Darryl Havens، الذي صمّم مكوّنات نظام التشغيل واستخدمها لأكثر من 12 سنة، برنامج إدارة الدخل / الخرج، والذي هو مكوّن موحّد لنظام الدخل / الخرج. يستعير الدخل / الخرج على النظام Windows NT بعض خصائصه من الأنظمة الأخرى التي عمل عليها Darryl ـ وبشكل خاص أنظمة التشغيل VAX/VMS وVAX ELN من DEC. وقد تطلّب دعم الأنظمة الفرعيّة Win 32 وOS/2 وPOSIX بعض المتطلّبات التي أثّرت على تصميم الدخل / الخرج.

تتضمّن الأهداف التصميميّة لنظام الدخل / الخرج ما يلي:

- توفير الدعم لأنظمة الملفّات المركّبة المتعدّدة بما فيها نظام الملفّات FAT ونظام ملفّات الأداء المرتفع (HPFS) ونظام ملفات CD-ROM (CDFS) ونظام ملفّات NT (NTFS) ونظام ملفّات قابل للإستعادة بالكامل جديد.
- توفير الخدمات لتسهيل تطوير مسيّق الجهاز قدر الإمكان وجعله يعمل على الأنظمة المتعدّدة المعالجات.
- الإتاحة لمدير النظام إضافة المسيقات إلى النظام أو إزالتها من النظام ديناميّاً.
- إحكام معالجة الدخل / الخرج مع إتاحة كتابة المسيقات في لغة عالية المستوى.
- توفير قدرات دخل / خرج ملف مخطّط لتنشيط الرسم وتجنّبه الملف وإستعمال التطبيق.

إضافة لهذه الأهداف المحدّدة، يجب أن يلبّي نظام الدخل / الخرج متطلّبات نظام التشغيل ككلّ. فمثلاً، يجب أن يكون نقّالاً ويجب أن يحمي موارده المشاركة بإستعمال الكائنات ويجب أن يوفّر القدرات لدعم تداخلات الدخل / الخرج مع Win 32 و OS/2 و POSIX ويجب أن يعمل بشكل صحيح على الأنظمة المتعدّدة المعالج.

يعالج هذا الفصل أولاً المزايا البنيويّة والتصميميّة لنظام الدخل / الخرج ثم يشرح كيفيّة معالجة طلبات الدخل / الخرج خلال تحرّكها في النظام. وهو يختتم بشرح النموذج المرتّب في طبقات والمستعمل لإنشاء المسيقات.

1-8 نظرة شاملة حول نظام الدخل / الخرج في NT:

نظام الدخل / الخرج في البرنامج التنفيذي NT هو مجموعة من شيفرات نظام التشغيل التي تقبل طلبات الدخل / الخرج من المعالجات في نمط المستعمل وفي نمط النواة ويسلّمها في شكل مختلف إلى أجهزة الدخل / الخرج. يوجد بين خدمات نمط المستعمل وميكانيكيّة عتاد الدخل / الخرج، عدّة مكوّنات للنظام سرّية، بما فيها أنظمة الملفّات الموسعة بالكامل ومسيّقات الأجهزة المتعدّدة ومسيق نقل شبكة واحد أو أكثر.

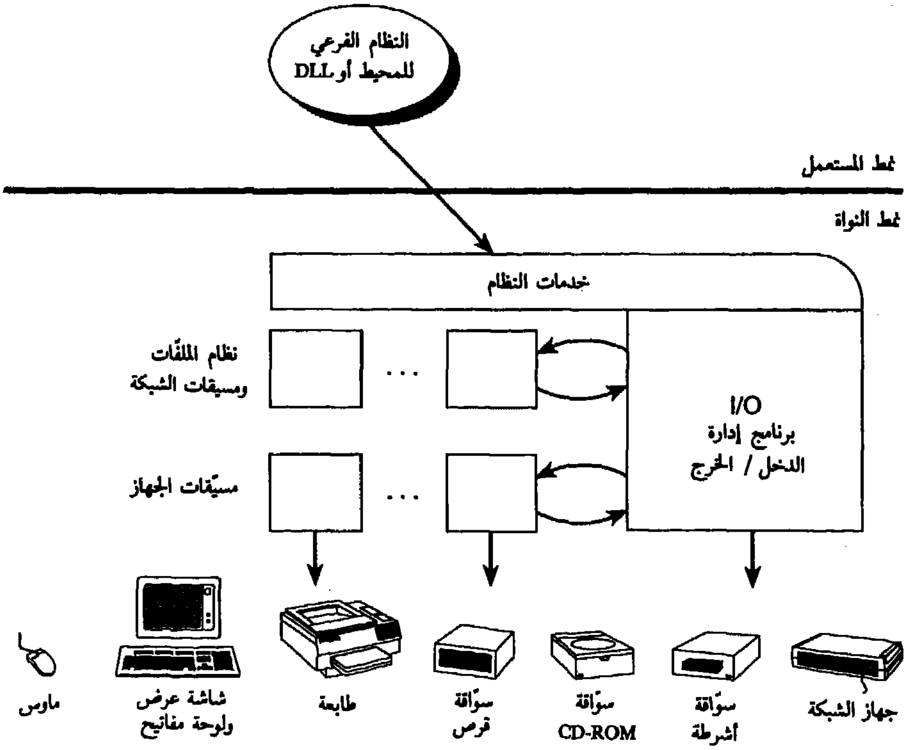
تبدأ النظرة الشاملة حول الدخل / الخرج في NT بتعريف مكوّنات نظام الدخل / الخرج وكيفيّة ملاءمتها سويّة. يلي ذلك، شرح يبرز تصميم نظام الدخل / الخرج: إستعماله للكائنات ونموذجه المتناسق لمسيقات الملفات ومسيقات الأجهزة وعمليات المزامنة وإشرافه على دخل / خرج الملفّات المخطّطة.

8-1-1 مكوّنات نظام الدخل / الخرج:

لاستيعاب تصميم نظام الدخل / الخرج في NT، يجب أولاً التعرُّف إلى الأجزاء المختلفة فيه. يوفّر الشكل (8-1) مشهداً مبسّطاً لبنية نظام الدخل / الخرج.

إن نظام الدخل / الخرج مدار بحزمة وهذا يعني أنه يمثّل كل طلب دخل / خرج بواسطة حزمة طلب دخل / خرج (IRP) عند تحرُّكها من مكوِّن نظام دخل / خرج واحد إلى آخر. وحزمة IRP هي بنية بيانات تتحكّم بكيفيّة معالجة عمليّة الدخل / الخرج في كل مرحلة خلال تحرُّكها.

يعرّف المكوّن الذي يسمى برنامج إدارة الدخل / الخرج إطار عمل منظّم ـ نموذج ـ حيث تسلّم ضمنه طلبات الدخل / الخرج إلى أنظمة الملفّات ومسيّقات الجهاز. ولا يدير برنامج



الشكل (8-1)
أجزاء نظام الدخل / الخرج

إدارة الدخل / الخرج فعلياً معالجة الدخل / الخرج. وتكون مهمّته إنشاء حزمة IRP تمثّل كل عمليّة دخل / خرج وتمرير IRP إلى المسيّق الصحيح والتخلّص من الحزمة عند إتمام عملية الدخل / الخرج. من الناحية المقابلة، يستلم مسيق حزمة IRP وينفّذ العمليّة التي تحدّدها الحزمة IRP وإما تمرّرها إلى برنامج إدارة الدخل / الخرج لإتمامها أو تمريرها إلى مسيق آخر (عبر برنامج إدارة الدخل / الخرج) لمعالجة إضافيّة.

في نظام الدخل / الخرج في NT، فالتعبير مسيّق يتّصف بمعنى أعمق من تعبير مسيّق الجهاز العام. فأنظمة الملفّات في NT هي مسيّقات «جهاز» معقّدة تقبل طلبات الدخل / الخرج إلى الملفّات وتلبّـي الطلبات عن طريق إصدار طلباتها الخاصة إلى مسيّقات الأجهزة الفعليّة. تتّصل مسيّقات نظام الملفّات ومسيّقات الجهاز عن طريق تمرير حزمات IRP.

إضافة لإنشاء حزمات IRP والتخلّص منها، يزوّد برنامج إدارة الدخل / الخرج الشيفرة العامة للمسيّقات المختلفة التي تستدعيها المسيّقات لتنفيذ معالجة الدخل / الخرج. وعن طريق دمج المهام العامة في برنامج إدارة الدخل / الخرج، تصبح المسيقات الإفراديّة أبسط ومرصوصة أكثر. فمثلاً، يوفّر برنامج إدارة الدخل / الخرج وظيفة تتيح لمسيّق واحد إستدعاء المسيّقات الأخرى. كذلك، فإنه يدير المخازن المؤقّتة لطلبات الدخل / الخرج ويوفّر دعم نفاذ الوقت للمسيّقات، ويسجّل أنظمة الملفّات القابلة للتركيب المحمّلة في نظام التشغيل.

كذلك، يوفّر برنامج إدارة الدخل / الخرج وسائل دخل / خرج مرنة تتيح للأنظمة الفرعيّة للمحيط، مثل Win 32 وPOSIX، إستخدام روتينات API للدخل / الخرج العائدة لها. وتتّصف الخدمات التي يوفّرها برنامج إدارة الدخل / الخرج بوظائفيّة تستطيع دعم متطلّبات الدخل / الخرج المختلفة في نمط المستعمل.

2-1-8 المزايا التصميميّة:

إن بنية نظام الدخل / الخرج في NT هي نتيجة الأهداف الإصطلاحيّة البارزة وتلك المستقبليّة. وعند إنشاء نظام تشغيل جديد، فالتوافقيّة الخلفيّة إعتبار مهمّ. وما لم يكن يصمّم نظام جديد ثوري دون تاريخ ــ حدث نادر ــ يجب أن يتوافق مع النظام، أو يشتغل داخلياً على الأقل، مع الأنظمة الموجودة. إن التوافقيّة الخلفيّة هي موضوع بارز بشكل خاص في نظام الدخل / الخرج، يطوّر مستعملو الحواسيب عادة البراجيّات أكثر من شراء عتاد جديد. ورغم أن نظام تشغيل جديد قد يتطلّب إضافة ذاكرة أو تخزين أقراص، فإنه نادراً ما يطلب من المستعمل شراء نموذج قرص صلب جديد أو تطوير الشاشة. وهذا يعني أنه يجب على نظام الدخل / الخرج دعم أجهزة الدخل / الخرج القديمة وحتى تلك العتيقة.

إن النظام Windows NT هو نظام تشغيل مصمَّم ليشتغل على معالجات حديثة مستعملاً تكنولوجياً التسعينات. ولقد كان ضبط متطلّبات الدخل / الخرج المتنوّعة إلى نموذج متجانس موحّد، إحدى التحديات في تصميم نظام الدخل / الخرج، وكان التحدي الآخر هو تصميم نموذج لا يمثّل أدنى قاسم مشترك للتكنولوجيا القديمة، لكن ذلك الذي يوفّر الحاجات المستقبليّة والذي يتشابك مع بقيّة نظام التشغيل. تقدّم الأقسام التالية بعض خصائص تعريف نموذج الدخل / الخرج في NT.

1-2-1-8 نموذج الكائن NT:

عندما تمّ تطويره أصلاً، قدّم نظام التشغيل UNIX نظرة مبسّطة جديدة للدخل / الخرج. واعتبرت كل البيانات المقروءة أو المكتوبة كمجرى بسيط من البايت الموجّهة إلى ملفّات ظاهريّة، والممثّلة بواسطة واصفات الملفّات. ينسب الملفّ الظاهري إلى أي مصدر أو مقصد للدخل / الخرج معتبر كأنه ملفّ. يحدّد نظام التشغيل لجهة كون الملفّ طرف كونسول أو أنبوب أو ملفّ حقيقي على قرص وهو يوجّه البيانات إلى الموقع الصحيح عند وقت التشغيل.

وفي Windows NT، تنفّذ البرامج أيضاً دخل / خرج على الملفّات الظاهريّة وتتناولها بإستعمال مقابض الملف. إن مبدأ مقبض الملفّ ليس جديداً، لكن ضمن البرنامج التنفيذي NT، ينسب مقبض الملفّ إلى كائنات الملفّ. تستدعي الشعَب في نمط المستعمل خدمات كائن الملفّ NT المحليّة للقراءة من ملفّ، والكتابة إلى ملفّ، وتنفيذ العمليّات الأخرى. يوجّه برنامج إدارة الدخل / الخرج ديناميأ طلبات الملفّ الظاهري هذه إلى الملفّات الفعليّة وإلى أدلّة الملفّات وإلى الأجهزة الفعليّة وإلى الأنابيب وإلى الشبكات وإلى الشقوق البريديّة وإلى أي مقاصد مدعومة في المستقبل.

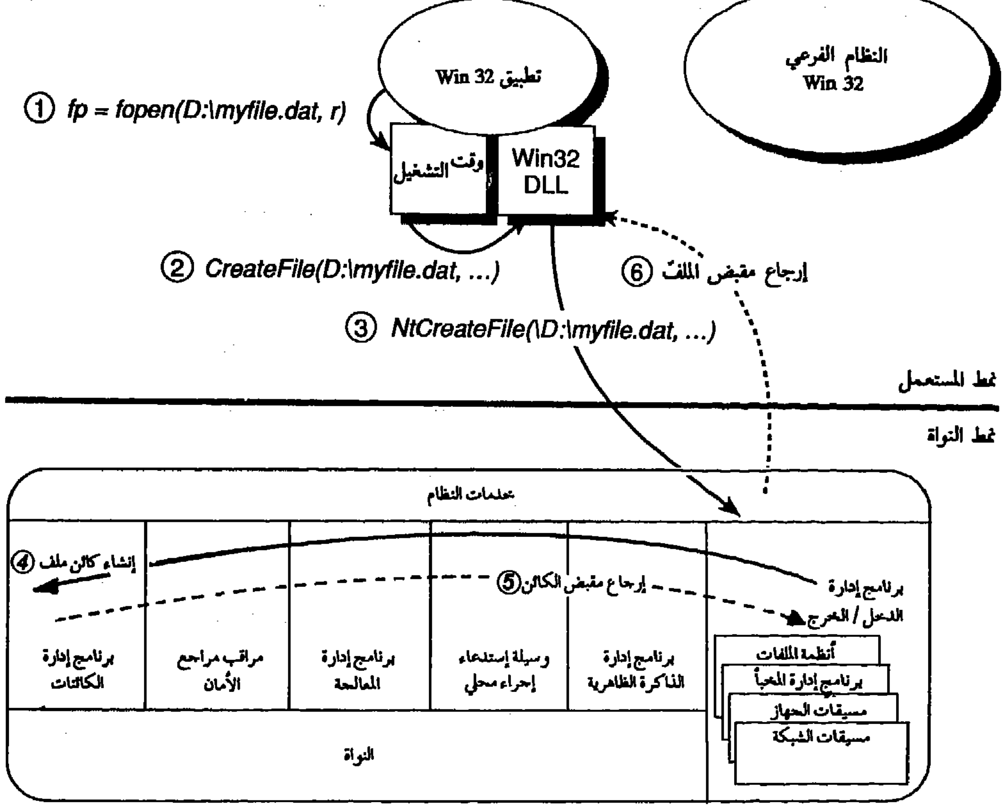
كما في أنظمة التشغيل الأخرى، يفتح تطبيق ملفّ بإستعمال وظيفة مكتبة قياسيّة في لغة برمجة مثل اللغة C. لكن العودة إلى التطبيق، في شكل واحد أو آخر، هي مقبض إلى كائن ملف برنامج تنفيذي NT. فمثلاً، عندما يستدعي تطبيق Win 32 الوظيفة Fopen ()، تستدعي مكتبة وقت التشغيل C روتين API CreateFile () للنظام Win 32 الذي يستدعي بدوره خدمة كائن دخل / خرج في NT. يفتح برنامج إدارة الدخل / الخرج كائن ملفّ ويرجع مقبض كائن إلى مكتبة وقت التشغيل C التي ترجعه إلى البرنامج التطبيقي، كما يبينّ في الشكل (8-2) على الصفحة التالية.

تتّخذ مصادر الدخل / الخرج ومقاصده شكل الكائنات لأنها تلائم معيار الكائنات في Windows NT: وهي موارد النظام التي يمكن مشاركتها من قبل الشعب في معالجتين في نمط

المستعمل أو أكثر. تتمّ حماية كائنات الملفّ كالكائنات الأخرى، التي تحتوي أسماء تسلسليّة، من قبل الأمان المعتمد على الكائن وتدعم المزامنة وتتناول من قبل خدمات الكائن.

عند فتح ملفّ، يزوّد المستعمل إسم الملفّ ونوع الوصول المطلوب ـــ عادة الوصول للقراءة والكتابة والإلحاق أو الحذف. يمرّر الطلب إلى نظام فرعي لمحيط (أو DLL) الذي يستدعي خدمة النظام NT. وهذا الأمر يبدأ عملية البحث عن إسم كائن في برنامج إدارة الكائنات. وكما سبق وشرح فيالفصل الثالث، يبدأ برنامج إدارة الكائنات البحث في فسحة عنوان الكائن ثم ينقّل التحكّم إلى برنامج إدارة الدخل / الخرج لإيجاد كائن الملفّ.

وكسائر كائنات البرنامج التنفيذي، تتمُّ حماية كائنات الملفّات من قبل واصف الأمان الذي يحتوي لائحة التحكُّم بالوصول (ACL). وعندما تفتح شعبة ملفّ، يراجع برنامج إدارة



الدخل / الخرج النظام الفرعي للأمان ليحدّد إذا كان روتين ACL للملفّات يتيح للمعالجة الوصول إلى الملفّ وفقاً لطلب شعبته. فإذا كان كذلك، يمنح برنامج إدارة الكائنات الوصول ويربط حقوق الوصول الممنوحة مع مقبض الملفّ الذي يرجعه. وإذا إحتاجت هذه الشعبة أو شعبة أخرى في المعالجة لتنفيذ عمليات إضافيّة غير محدّدة في الطلب الأصلي، يجب عليها فتح مقبض آخر الذي يحثّ عملية تدقيق بالأمان أخرى. (راجع الفصل الثالث، «برنامج إدارة الكائنات وأمان الكائنات»، لمزيد من المعلومات حول حماية الكائن).

تستعمل أيضاً كائنات الملفّ للمزامنة. فبعد إصدار طلب دخل / خرج، تنتظر الشعبة مقبض ملفّ لمزامنة تنفيذه مع إتمام سوّاقة القرص أو جهاز آخر عمليّة نقل البيانات. تتعلّق قدرة المزامنة هذه مع المزيّة الهامة الأخرى لنظام الدخل / الخرج ـ وهي عمليّات الدخل / الخرج غير المتزامنة.

2-2-1-8 نموذج المسيّق المتناسق:

الخاصية الثامنة لنظام الدخل / الخرج هي البنية المتناسقة لمسيّقاته والتعريف الواسع لما يتألّف منه المسيق. ففي البرنامج التنفيذي NT، يبيّن مسيّق الجهاز ونظام الملفّات بنفس الطريقة ويمثّلان شكلاً مطابقاً لبقيّة نظام التشغيل. إضافة لذلك، تعتبر الأنابيب المسمّاة ومعيدات توجيه الشبكة (البراعجيات التي توجّه طلبات الملفّات على عدّة شبكات متنوّعة) على أنها «أنظمة ملفّات» وتستخدم كمسيّقات لنظام الملفّات. وكل مسيق هو مكوّن ذاتي الإحتواء يمكن إضافته إلى نظام تشغيل أو نزعه منه دينامياً.

يعرّف برنامج إدارة الدخل / الخرج نموذج تنشأ حوله المسيّقات. وتتضمّن الخصائص الرئيسيّة لنموذج المسيق ما يلي:

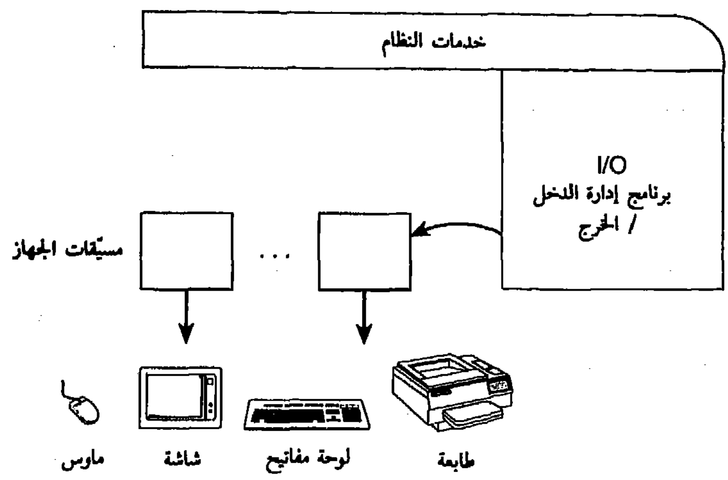
- المسيّقات نقّالة ويمكن كتابتها في لغة عالية المستوى. وهي مصمّمة بحيث تحتاج للقليل من التغييرات أو لا تغييرات بتاتاً من تصميم معالج إلى آخر. ولا تحتاج المسيّقات من المستوى الممتاز؛ مثل أنظمة الملفّات، إلى أي تغيير.
- عمليات الدخل / الخرج مدارة بحزمة، ومنظّمة حول إرسال روتينات IRP من مسيّق إلى آخر. ويمكن معاودة إستعمال روتينات IRP خلال مرورها عبر الطبقات المختلفة لنظام الدخل / الخرج.
- يستطيع نظام الدخل / الخرج تعيين المسيّقات للتحكّم بأجهزة إضافيّة أو مختلفة إذا تغيّر تشكيل النظام.

■ يجب أن تزامن المسيّقات وصولها إلى بيانات المسيّق العامة. ويمكن شفع تنفيذ مسيّق من قبل الشعب بأولويّة أعلى ما يمكن مقاطعته من قبل مقاطعات بأولويّة أعلى. وهذا الواقع، إضافة إلى قدرة NT على تشغيل شيفرة المسيّق في نفس الوقت على أكثر من معالج واحد في حاسوب متعدّد المعالجات، يتطلّب مراعاة خاصة لمسألة المزامنة.

■ يجب أن تستعاد المسيقات بعد إنقطاع الطاقة الكهربائيّة وتعيد بدء عمليّات الدخل / الخرج المقاطعة.

يتيح التداخل المنظومي المتناسق الذي تمثّله المسيّقات لبرنامج إدارة الدخل / الخرج إستدعاء أي مسيّق يريده دون الحاجة لأية معرفة خاصة ببنيته أوتفاصيله الداخليّة. تستطيع المسيّقات أيضاً إستدعاء بعضها البعض (عبر برنامج إدارة الدخل / الخرج) لتحقيق معالجة مستقلّة مرتّبة في طبقات لطلب دخل / خرج.

راجع المثال التالي. يقبل نظام ملفّات طلب قراءة الأحرف من ملفّ معيّن. ويترجم الطلب إلى طلب بدء القراءة من القرص عند موقع «منطقي» معيّن ويتابع القراءة لعدد معيّن من البايت. ويمرّر هذا الطلب إلى مسيّق قرص عادي. يترجم مسيّق القرص، بدوره، الطلب إلى موقع إسطوانة / مسار / مقطع على القرص ويستعمل رؤوس القرص لإسترداد البيانات. تتيح قدرة وضع المسيّقات على بعضها البعض في طبقات بهذه الطريقة للمسيّقات أن تكون منظومة وتزيد من معاودة إستخدام شيفرة المسيّق.



الشكل (3-8)
دخل / خرج إلى مسيّق آحادي الطبقة

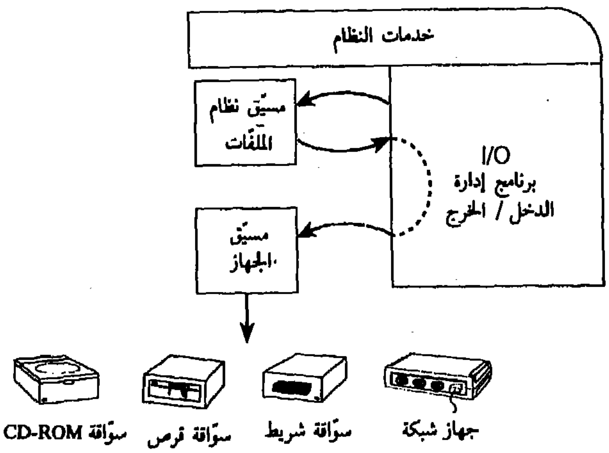
يستطيع نموذج الدخل / الخرج المرتّب في طبقات إستيعاب عدد غير محدّد من المسيّقات. وبواسطة المسيّقات الآحاديّة الطبقة، يمرّر طلب دخل / خرج إلى برنامج إدارة الدخل / الخرج ثم إلى مسيّق الجهاز الذي يتّصل مباشرة مع الجهاز، على النحو المبيّن في الشكل (8-3).

يظهر الشكل (8-4) مثال عن مسيّق متعدّد الطبقات حيث يمرّ طلب عبر مسيّقين أو أكثر خلال معالجته. في هذا المثال، فإن الطبقتين هما مسيق نظام الملفّات ومسيّق جهاز. لاحظ أن المسيّق بمستوى أعلى لا يستدعي مسيّقاً بمستوى أدنى مباشرة، لكنّه يستدعى برنامج إدارة الدخل / الخرج الذي يستدعي المسيّق بمستوى أدنى.

إن المسيّقات المتعدّدة الطبقات أكثر إستعمالاً من المسيّقات الآحاديّة الطبقة، رغم أنه يمكن الوصول إلى بعض الأجهزة العاملة بالبايت، مثل الأجهزة التسلسليّة أو المتوازية بإستعمال مسيّق جهاز آحادي الطبقة.

يتمّ الوصول دائماً إلى أجهزة التخزين الكبيرة بإستعمال مسيّقات متعدّدة الطبقات. فالطلب يمرّ أولاً عبر مسيّق نظام ملفّات ثم عبر مسيّق جهاز.

يمكن أيضاً إنشاء مجموعات أكثر تعقيداً من المسيّقات المرتّبة في طبقات. فمثلاً، يمكن أن يحتوي حاسوب أجهزة متعدّدة مثل سوّاقات قرص أو شريط مثبّتة بالناقل العمومي SCSI (أي، نظام التداخل مع نظام الحاسوب الصغير). وقد يمرّ طلب دخل / خرج إلى سوّاقة قرص كهذه عبر المسيّقات التالية:



الشكل (8-4)
دخل / خرج إلى مسيّق متعدّد الطبقات

- مسيّق نظام الملفّات.
- مسيّق فئة قرص يصدر طلبات SCSI.
- مسيّق منفّذ SCSI يرسل الطلبات إلى القرص بإستعمال بروتوكول الناقل العمومي SCSI.

إن كل من هذه المسيّقات منظومي بحيث يمكن إستعمالها جميعاً في تشكيلات أخرى أيضاً.

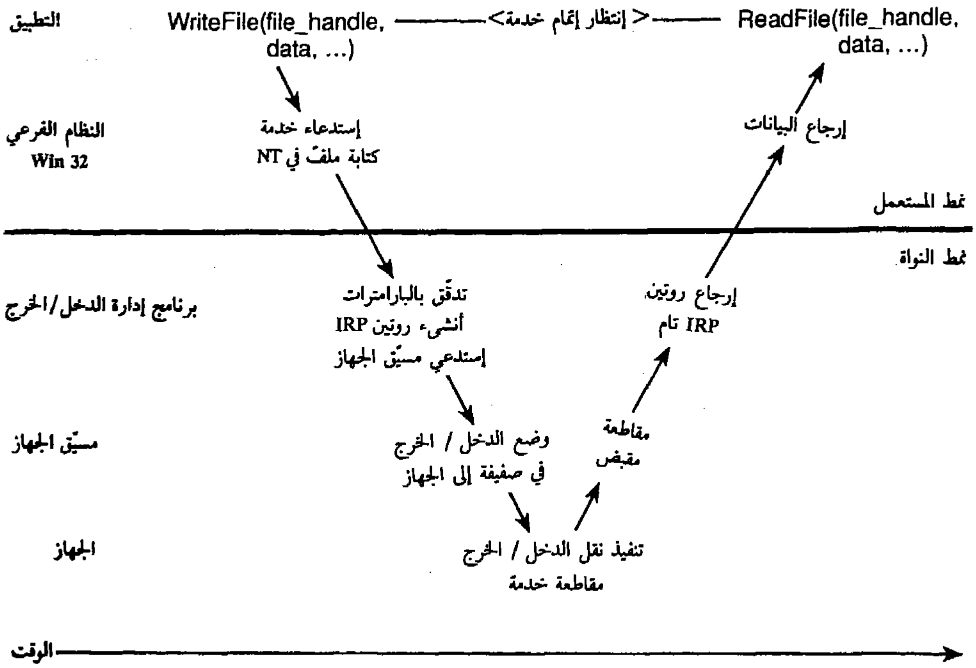
3-2-1-8 العمليّة غير المتزامنة:

المزيّة الثالثة لنظام الدخل / الخرج في NT هي طبيعته اللاتزامنيّة. يعرّف الدخل / الخرج غير المتزامن بسهولة عن طريق شرح مقابلة: أي الدخل / الخرج المتزامن. ومعظم المبرمجين يعرفون الدخل / الخرج المتزامن عند إستدعاء خدمة دخل / خرج، يتمّم الجهاز نقل البيانات ثم يرجع شيفرة الحالة إلى البرنامج، ويستطيع البرنامج الوصول إلى البيانات المنقولة فوراً. وعند إستعمالها في أبسط أشكالها، تنفّذ بتزامن روتينات ReadFile () API و WriteFile () في Win 32 على سبيل المثال. وهي تنتهي عملية دخل / خرج قبل إرجاع التحكُّم إلى المستدعي، كما يوضح الشكل (8-5) على الصفحة التالية.

الدخل / الخرج المتزامن هو قياسي على معظم أنظمة التشغيل وهو مناسب لمعظم الحالات. لكن المعالجات الحديثة سريعة جداً ـ أسرع من معظم أجهزة الدخل / الخرج. فخلال معالجة جهاز طلب دخل / خرج واحد، يستطيع المعالج تنفيذ الآلاف من أسطر الشيفرة. عادة، يجب أن يتمكّن التطبيق من إستعمال المعالج خلال نقل الجهاز للبيانات. لهذا السبب، يوفّر برنامج إدارة الدخل / الخرج في NT قدرات دخل / خرج غير متزامنة. يحدّد نظام فرعي إستعمال دخل / خرج متزامن أو غير متزامن ووفقاً لكيفيّة إشتغال روتين API، فإنه يستطيع توفير أي نوع من الدخل / الخرج إلى تطبيقات المستضاف. يوفّر النظام الفرعي Win 32 روتينات API للوصول إلى الملف التي يمكن تنفيذها إما تزامنياً أو غير تزامني.

تتيح الخدمات غير المتزامنة لتطبيق إصدار طلب دخل / خرج ثم مواصلة التنفيذ خلال نقل الجهاز للبيانات، كما يظهر ذلك في الشكل (8-5).

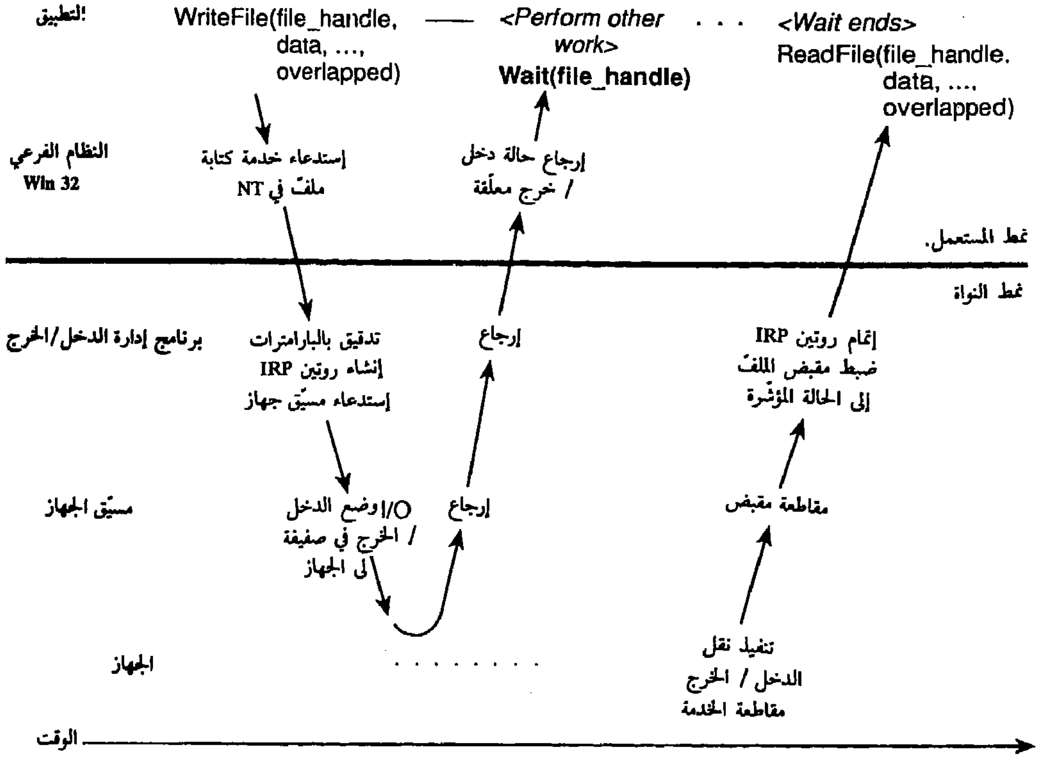
يوفّر الدخل / الخرج غير المتزامن ميّزة مهمة على الدخل / الخرج المتزامن، وهي تحسين سرعة تنفيذ التطبيق. فخلال إنشغال الجهاز في نقل البيانات، يستمرُّ التطبيق تنفيذ العمل الآخر. فمثلاً، يستطيع التطبيق كتابة رسم إلى شاشة خلال تعبئة مسيّق جهاز لمخزن مؤقّت



الشكل (8-5)
الدخل / الخرج المتزامن

بالبيانات من ملف إلى قرص. لإستعمال الدخل / الخرج غير المتزامن، يجب أن تحدّد الشعبة دخل / خرج غير متزامن (متراكبة في مصطلح Win 32) عندما تفتح مقبضاً. بعد إصدار عمليات دخل / خرج غير متزامنة، يجب أن تكون الشعبة حذرة لجهة عدم الوصول إلى أي بيانات من عمليّة الدخل / الخرج إلى أن ينتهي مسيّق الجهاز من نقل البيانات. بمعنى آخر، يجب أن تزامن الشعبة تنفيذها مع إتمام طلب الدخل / الخرج عن طريق إنتظار مقبض، كما يظهر في الشكل (8-6).

إن حوالي ثلث خدمات NT المحليّة التي يوفّرها برنامج إدارة الدخل / الخرج إلى الأنظمة الفرعيّة ومكتبات DLL غير متزامنة إفتراضياً. إن الخدمات غير المتزامنة هي التي يحتمل أن تكون عمليات طويلة أو بطول غير متوقّع ـ فمثلاً، قراءة ملفّ أو كتابته أو سرد محتويات دليل ملفّ. يجب على الشعبة التي تستدع هذه الخدمات أن تزامن تنفيذها مع إتمامها. وبشكل بديل، يستطيع مستدعي جعل كل خدمات NT تتصرّف بتزامن عن طريق تحديد الدخل / الخرج عند فتح مقبض ملفّ.



الشكل (8-6)

الدخل / الخرج غير المتزامن

يوجد فرق بين كيفيّة تصرُّف خدمة لمستدعيها وكيفيّة إستخدامها فعلياً من قبل نظام الدخل / الخرج في NT. رغم أن بعض الخدمات تتصرّف تزامنياً والآخر غير تزامني، فإن نظام الدخل / الخرج يعمل في غير تزامن كلياً ـــ من معالجة المقاطعة إلى تمرير النتائج مجدداً إلى نمط المستعمل لبدء طلبات الدخل / الخرج على جهاز. يوفّر العمل غير المتزامن لنظام الدخل / الخرج مرونة قصوى في تنفيذ المهام الأخرى خلال عمليّة نقل البيانات من قبل الأجهزة الأبطأ نسبياً. يتم وصف إستدعاءات الإجراء غير المتزامن (APCs)، وهي مزيّة أخرى للنظام NT والدخل / الخرج غير المتزامن في Win 32، في القسم 8-2-2-3.

8-1-2-4 دخل / خرج الملفّ المخطّط وتخبئة الملفّ:

إحدى المزايا المهمّة لنظام الدخل / الخرج هي دخل / خرج الملفّ المخطّط، الذي يصدر بجهد مشترك من قبل نظام الدخل / الخرج وبرنامج إدارة الذاكرة الظاهريّة (VM). وضمن

نظام التشغيل، يستعمل دخل / خرج الملفّ المخطّط للوظائف المهمّة مثل تخبئة الملفّات وتنشيط الرسم (تحميل برامج قابلة للتنفيذ وتشغيلها). كذلك، يوفّر برنامج إدارة VM دخل / خرج الملفّ المخطّط إلى نمط المستعمل عبر الخدمات المحليّة. تستطيع الأنظمة الفرعيّة للمحيط إستعمال الخدمات لتوفير قدرات الملفّ المخطّط إلى تطبيقات المستضاف.

ينسب دخل / خرج الملفّ المخطّط إلى قدرة مشاهدة ملفّ يستقرّ على قرص كجزء من ذاكرة ظاهريّة لمعالجة. يستطيع البرنامج الوصول إلى الملفّ كصفيفة كبيرة دون تخزين البيانات مؤقّتاً أو تنفيذ دخل / خرج قرص. يتمكّن البرنامج من الوصول إلى الذاكرة، ويستعمل برنامج إدارة VM آلية الترتيب في صفحات لتحميل الصفحة الصحيحة من ملفّ القرص. وإذا كتب التطبيق إلى فسحة العنوان الظاهري، يكتب برنامج إدارة VM التغييرات على الملفّ كجزء من عمليّة الترتيب في الصفحات العاديّة.

يمكن للتطبيقات التي تنفّذ الكثير من عمليات الدخل / الخرج إلى ملفّ أو تلك التي تتمكّن من الوصول إلى أجزاء من الملفّات المختلفة المتعدّدة، أن تسرّع تنفيذها عن طريق إستعمال دخل / خرج مخطّط لأن الكتابة إلى الذاكرة أسرع بكثير من الكتابة إلى جهاز. كذلك، يستمثل برنامج إدارة VM وصوله إلى القرص، بحيث يتيح الدخل / الخرج المخطّط للتطبيقات الإستفادة من خبراته.

يستعمل مكوّن نظام الدخل / الخرج برنامج إدارة المخبأ، الدخل / الخرج المخطّط لإدارة المخبأ المعتمد على الذاكرة. تستعمل أنظمة الملفّات وملقّم شبكة النظام Windows NT المخبأ لوضع بيانات الملفّ التي يتم الوصول إليها بتكرار في الذاكرة لتوفير وقت إستجابة أفضل للبرامج المتعلّقة بالدخل / الخرج. وبينما تحدّد أنظمة التخبئة عدداً ثابتاً من البايت لتخبئة الملفّات في الذاكرة، يكبر مخبأ NT أو يصغر وفقاً لكميّة الذاكرة المتوفّرة. وعندما تفتح شعبة وتستعمل ملفاً، يطلب نظام الملفّات من برنامج إدارة المخبأ إنشاء كائن قسم دون إسم وتخطيط ملفّ المستدعي فيه. وعندما يستعمل المستدعي الملفّ يجلب برنامج إدارة VM الصفحات التي تمّ الوصول إليها إلى كائن القسم من القرص ويرجعها إلى القرص خلال عمليّة الترتيب في صفحات. يوسّع ناقل الصفحات تلقائياً حجم المخبأ (بإستعمال آليات ضبط العمل العادي) عند توفّر كميّة وافرة من الذاكرة ويصغّر المخبأ عندما يحتاج لصفحات خالية. وبالإستفادة من نظام الترتيب في صفحات لبرنامج إدارة VM، يتجنّب برنامج إدارة المخبأ إستنساخ العمل الذي سبق ونفّذه برنامج إدارة VM.

2-8 معالجة الدخل / الخرج:

لقد وصف القسم السابق نظام الدخل / الخرج في NT من النواحي الخارجيّة حيث ركّز على مزاياه التصميميّة المتنوّعة. أما الخطوة التالية لإستيعاب الدخل / الخرج في المستوى التنفيذي فهي في شرح النواحي الداخليّة لنظام الدخل / الخرج. ولأن روتينات IRP تقوم بذلك، يتعامل هذا القسم مع عدّة روتينات IRP خلال تحرُّكها في النظام.

تمرُّ طلبات الدخل / الخرج عبر عدّة مراحل متوقّعة من المعالجة. تتغيّر المراحل وفقاً لكون الطلب عائد لجهاز مشغّل بواسطة مسيّق آحادي الطبقة أو جهاز بمسيّق متعدّد الطبقات. وتختلف المعالجة أيضاً وفقاً لتحديد المستدعي لدخل / خرج متزامن أو غير متزامن.

تبدأ معظم طلبات الدخل / الخرج بنفس الطريقة. وبعد فتح مقبض ملفّ، يستدعي تطبيق روتين دخل / خرج. يُزوَّد عادة الروتين من قبل مكتبة لغة أو نظام فرعي لمحيط. فمثلاً، يستطيع مبرمج Win 32 إستدعاء روتين () C function read أو إستدعاء روتين ReadFile () API في Win 32 . في أية حالة، يستدعي النظام الفرعي Win 32 (أو مكتبة DLL) خدمة نظام دخل / خرج محليّة.

رغم تقديم الأنظمة الفرعيّة لمحيط مقابض الملفّ في عدّة طرق مختلفة، تحتوي معظم مقابض ملفّ في نمط المستعمل على مقبض كائن NT كقلبها. وتقدّم الملفّات في NT ككائنات، ويزوّد نظام الدخل / الخرج بوصف كائنات الملفّ NT وخدمات كائن الملفّ المحليّة. ويصف القسم الثاني ماذا يحدث عندما يحفز نظام دخل / خرج في NT، بإستعمال مثال طلب إلى جهاز مدار بمقاطعة محكوم بواسطة مسيّق أحادي الطبقة. يوسّع القسم الثالث شرح المسيّقات المتعدّدة الطبقات ويظهر طلب دخل / خرج وهو يمرُّ عبر أكثر من مسيّق واحد قبل إتمامه. ويشرح القسم الأخير مسائل البرمجة التي تحيط إستعمال خدمات دخل / خرج غير متزامنة.

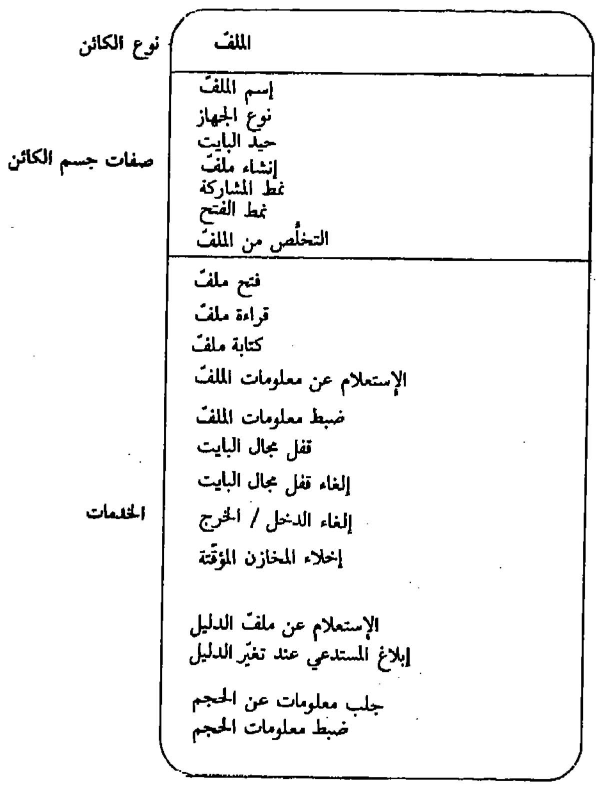
1-2-8 كائنات الملفّ:

رغم كون معظم الموارد المشاركة في Windows NT موارد تعتمد على الذاكرة، فإن معظم تلك التي يديرها نظام الدخل / الخرج إما موجودة على أجهزة أو إنها أجهزة فعليّة. ورغم هذا الإختلاف، فإن الموارد المشاركة في نظام الدخل / الخرج، كتلك في المكوّنات الأخرى في البرنامج التنفيذي NT، تتناول ككائنات.

تتطلّب الكائنات المداورة بواسطة نظام الدخل / الخرج مناولة خاصة، لأن برنامج إدارة الكائنات في NT لا يعرف الكثير حول بنيات دليل نظام الملفّات ويعرف أقل من ذلك حول بنية

الموجود على القرص أو نسق البيانات المخزّنة على شريط. وبسبب هذه الأسباب والتعقيدات الأخرى الناتجة عن الأجهزة الفعليّة، فإن إحدى الصعوبات الرئيسيّة في إنشاء نظام الدخل / الخرج هو تكامله في نظام الكائنات.

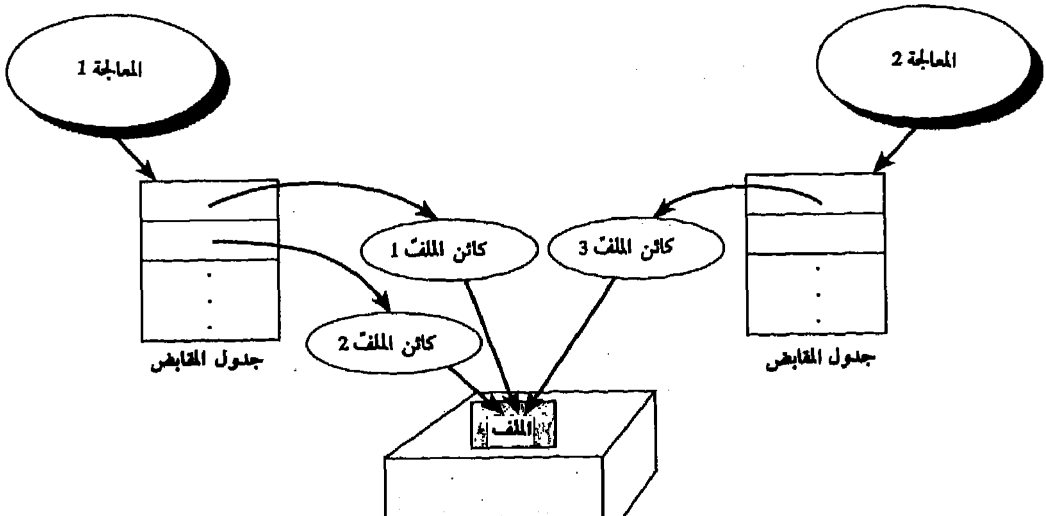
إن كائنات الملفّ هي الوسيط. فهي توفّر عرضاً يعتمد على ذاكرة من الموارد الفعليّة المشاركة. وعندما يفتح مستدعي ملفّ أو جهاز بسيط، يرجع برنامج إدارة الدخل / الخرج مقبضاً إلى كائن ملفّ NT. يعامل برنامج إدارة الكائنات كائنات الملفّات كأي كائن آخر إلى أن يحتاج لإسترداد بعض المعلومات من جهاز أو يخزّن معلومات على جهاز، فعند ذلك يستدعي برنامج إدارة الكائن برنامج إدارة الدخل / الخرج لمساعدته في الوصول إلى الجهاز. (راجع الفصل الثالث. «برنامج إدارة الكائنات وأمان الكائن» لمزيد من المعلومات). يلخّص الشكل (8-7) محتويات كائنات الملفّ والخدمات التي تعمل عليها (وبالتالي، على الملفّات المفتوحة أو الأجهزة التي تعرضها). ويصف الجدول (8-1) صفات كائن الملفّات.



الشكل (8-7)
كائن الملفّ

الصفة	الغرض
إسم الملفّ	يعرِّف الملفّ الفعلي الذي يعود إليه كائن الملفّ
نوع الجهاز	يشير إلى نوع الجهاز حيث يستقرُّ الملفّ
حيد البايت	يعرِّف الموقع الحالي في الملفّ (يصلح فقط للدخل / الخرج المتزامن)
نمط المشاركة	تشير إلى إمكانيّة فتح مستدعي آخر الملفّ لعمليات القراءة والكتابة أو الحذف خلال إستعمال من هذا المستدعي نمط الفتح يشير إلى كون الدخل / الخرج متزامن أو غير متزامن، مخبأ أو غير مخبأ، تسلسلي أو عشوائي وما شابه
التخلّص من الملفّ	يشير إلى إمكانيّة حذف الملفّ بعد إغلاقه

الجدول (8-1)
صفات كائن الملفّ



الشكل (8-8)
مشاركة ملفّ

لأن كائن الملفّ هو عرض يعتمد على ذاكرة لمورد مشارك وليس المورد نفسه، فإنه يختلف عن كائنات البرنامج التنفيذي الأخرى. فكائن الملفّ يحتوي فقط على البيانات الفريدة لمقبض

كائن، بينما يحتوي الملفّ نفسه البيانات أو النصّ الواجب مشاركته. وفي كل مرة تفتح شعبة مقبض ملفّ، يتمّ إنشاء كائن ملفّ جديد مع مجموعة جديدة من الصفات الخاصة بمقبض. فمثلاً، ينسب حيد بايت الصفة إلى موقع في الملفّ حيث ستحصل عمليّة القراءة أو الكتابة التالية بإستعمال ذلك المقبض. تحتوي كل شعبة التي تفتح مقبضاً إلى ملفّ على حيد بايت خاص حتى إذا كان الملفّ المحدّد مشارك. في الواقع، يمكن إعتبار صفات كائن الملفّ كأنها خاصة بمقبض واحد كما يوضح الشكل (8-8).

رغم كون مقبض ملفّ خاص بمعالجة، غير أن المورد الفعلي المحدّد ليس كذلك. لهذا، وكما عند إستعمال أي مورد مشارك، يجب أن تزامن الشعَب وصولها إلى الملفّات المشاركة، وأدلّة الملفّات أو الأجهزة. وإذا كانت شعبة تكتب إلى ملفّ، على سبيل المثال، يجب أن تحدّد منع الوصول إلى الكتابة عند فتح مقبض الملفّ لمنع الشعَب الأخرى من الكتابة إلى الملفّ في نفس الوقت. وبشكل بديل، يمكنه قفل أجزاء من الملفّ خلال الكتابة إليه.

عندما يشير مقبض ملفّ إلى ملفّ فعلي (عكس ما يشير إلى جهاز أو أنبوب أو «ملفّ» آخر)، تستطيع الشعبة إستعمال مقبض الملفّ لإسترداد المعلومات المخزّنة إما في كائن الملفّ أو في الملفّ نفسه. يوجد في الجدول (8-2) بعض المعلومات المخزّنة في ملفّ (أو دليل ملفّ، حيث أمكن) إضافة إلى البيانات المستقرّة في كائن ملفّ. تتنوّع هذه المعلومات مع أنظمة الملفّات المختلفة. كذلك، تعرّف الأنظمة NTFS و HPFS الصفات الممدّدة.

الصفة	الغرض
وقت الإنشاء	تشير إلى تاريخ ووقت إنشاء الملفّ
وقت آخر وصول	تشير إلى تاريخ ووقت آخر مرّة قرأ الملفّ أو كتب إليه
وقت آخر كتابة	تشير إلى تاريخ ووقت آخر تغيير في الملفّ
وقت آخر تغيير في الوقت	تشير إلى تاريخ ووقت آخر مرّة غيّر الوقت
خصائص الملفّ	تشير إلى كون الملفّ ملفّ قراءة فقط أو ملفّ نظام أو ملفاً مخفياً أو ملفّ أرشيف أو ملفّ تحكّم
حجم التحصيص	تشير إلى حجم حصة الملفّ بالبايت
نهاية ملفّ	تعلّم حيد أول بايت حرّ في الملفّ

الجدول (8-2)
صفات الملفّ

8-2-2 طلب دخل / خرج إلى مسيّق أحادي الطبقة:

إن وظيفة برنامج إدارة الدخل / الخرج هي قبول طلب دخل / خرج، وإستعمال مقبض الملفّ المزوّد لمعالجة طلب الدخل / الخرج وإرسال النتيجة إلى المستدعي. ولتوضيح معالجة طلبات الدخل / الخرج في البرنامج التنفيذي NT، يتناول هذا القسم مسار روتين IRP إلى داخل نظام الدخل / الخرج وإلى خارجه. وفي المثال الذي يلي، يصدر مستدعي في نمط المستعمل مثل نظام فرعي لمحيط أو DLL، طلب متزامن إلى جهاز مدار بمقاطعة بسيط. وهذا الجهاز محكوم بمسيّق آحادي الطبقة.

تتابع معالجة طلب متزامن في ثلاث مراحل:

1 ـــ يرسل برنامج إدارة الدخل / الخرج الطلب على شكل روتين IRP إلى المسيّق (في هذه الحالة، مسيّق جهاز) ويبدأ المسيّق عمليّة الدخل / الخرج.

2 ـــ يتمّم الجهاز العمليّة ويقاطع، ويخدم مسيّق الجهاز المقاطعة.

3 ـــ يتمّم برنامج إدارة الدخل / الخرج طلب الدخل / الخرج.

وفي المثال الثاني الذي يلي، يصدر مستدعي في نمط المستعمل طلب دخل / خرج غير متزامن. تختلف معالجة طلب غير متزامن عن معالجة طلب متزامن من ناحية واحدة مبدئياً. فالإستدعاء غير المتزامن يضيف خطوة بين الخطوتين 1 و2، حيث يقوم برنامج إدارة الدخل / الخرج بإرجاع التحكّم إلى المستدعي. يمكن بعد ذلك أن يواصل المستدعي أعمالاً أخرى خلال متابعة الخطوتين 2 و3، لكن عليه المزامنة مع إتمام الخطوة 3 لكي يعرف متى تمّ نقل البيانات. تعرض بتفصيل المراحل الثلاث لمعالجة طلبات الدخل / الخرج المتزامنة وغير المتزامنة في الأقسام التالية.

1-2-2-8 وضع طلب دخل / خرج في صفيفة:

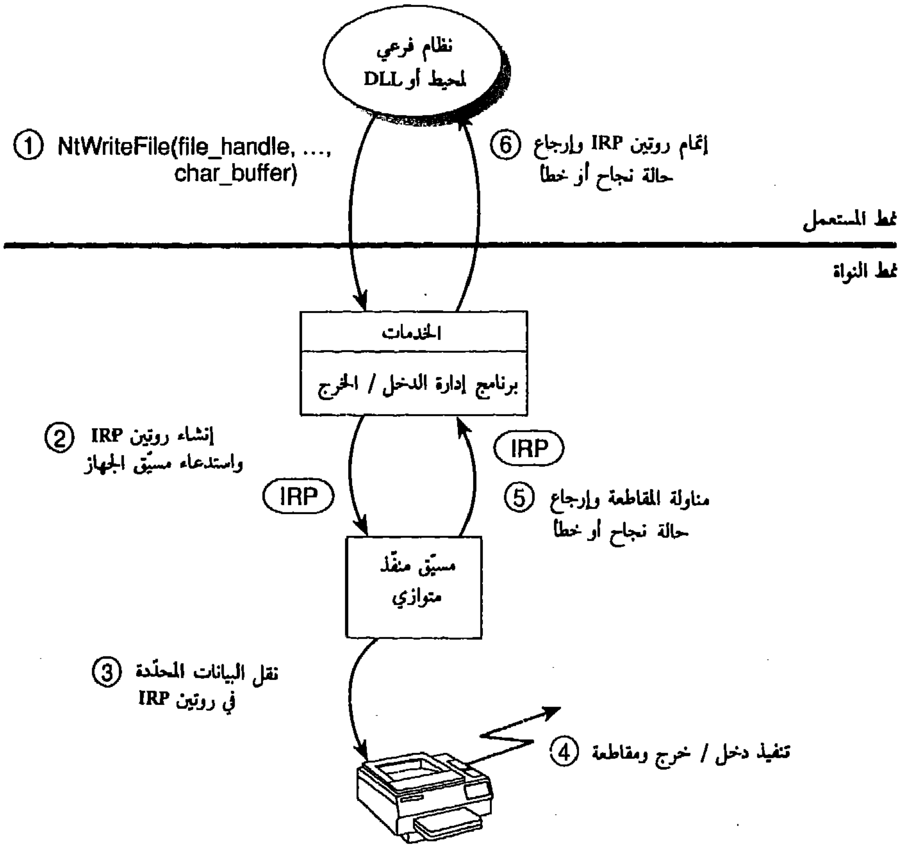
لنبدأ بمثال بسيط. إفترض أن تطبيقاً يكتب بتزامن مخزّن مؤقّت للأحرف إلى طابعة. والطابعة مثبّتة إلى المنفّذ المتوازي للحاسوب، وهي تعمل بواسطة مسيّق منفّذ متوازي آحادي الطبقة. (عادة، تخزّن طلبات الطابعة في القرص أولاً، لكن بغية التبسيط، يتجاهل هذا المثال تلك الخطوة).

في النظام Windows NT، يمرُّ طلب الطابعة أولاً عبر نظام فرعي لمحيط أو DLL الذي يستدعي بدوره الخدمة () NtWriteFile لبرنامج إدارة الدخل / الخرج. ويكون أول بارامتر

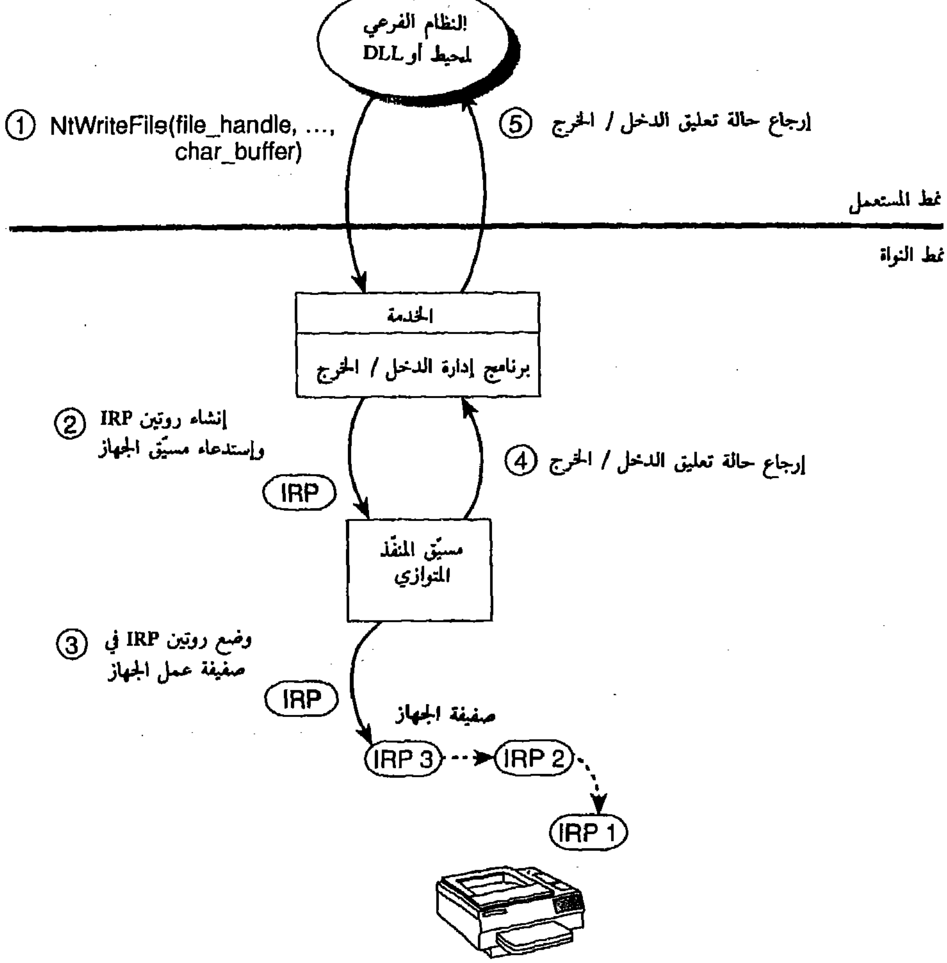
للخدمة () NtWriteFile وهو مقبض إلى كائن ملفّ، الذي يمثّل مقصد طلب الدخل / الخرج. ولأن المقصد هو منفذ متوازي، يجب أن يكون النظام الفرعي قد قام بفتح مقبض إلى المنفذ (ملفّ ظاهري يُعرَف بإسم Device\Parallel 0\) وتحديد دخل / خرج متزامن.

ينشىء برنامج إدارة الدخل / الخرج روتين IRP حيث يخزّن فيه مؤشّراً إلى كائن الملفّ وشيفرة وظيفة تبلغ مسيّق المنفّذ المتوازي عن العمليّة الواجب تنفيذها ـ في هذه الحالة، عمليّة كتابة. يحدّد برنامج إدارة الدخل / الخرج موقع المسيّق ثم يستدعيه حيث يمرّر روتين IRP إلى المسيّق. يوضح الشكل (8-9) المسار الذي يسلكه روتين IRP من هناك.

ففي الشكل (8-9)، تمثّل الخطوات 1 إلى 3 وضع طلب الدخل / الخرج المتزامن في



الشكل (8-9)
وضع طلب متزامن في صفيفة وإتمامه



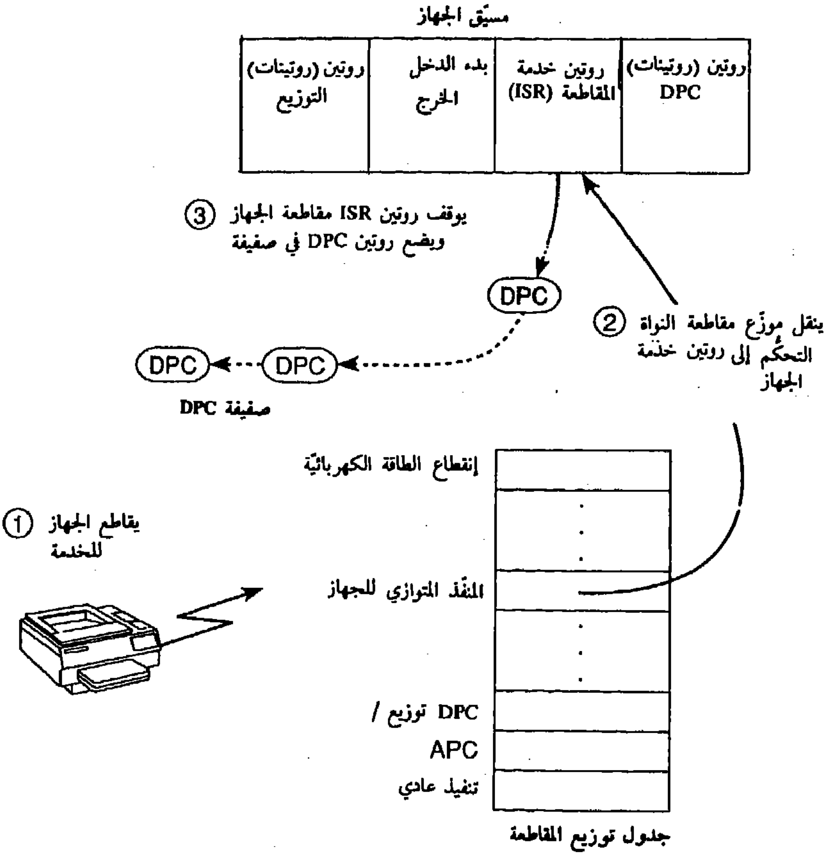
الشكل (10-8)
وضع طلب غير متزامن في صفيفة

صفيفة. وتبينّ الخطوة 5، التي هي خدمة المقاطعة، والخطوة 6، والتي هي إتمام طلب الدخل / الخرج في شكل مختصر لتوضيح إنسياب روتين IRP. وهي تشرح بتفصيل أكبر في الأقسام اللاحقة.

أما طلب الدخل / الخرج غير المتزامن، فإنه يتابع بطريقة مختلفة قليلاً، كما يوضح ذلك الشكل (10-8).

ففي الشكل (10-8)، فإن الطابعة مشغولة بعدّة طلبات منتظرة. وعلى الطلب الجديد أن

ينتظر دوره. يضع مسيّق الجهاز روتين IRP في صفيفة خاصة بالجهاز ويرجع فوراً الحالة «تعليق الدخل / الخرج» التي ترجع إلى المستدعي. ويستطيع التطبيق (أو النظام الفرعي) مواصلة عمله مع إتّباع المؤشّر ببطء، فعلى سبيل المثال، قد يحضر التطبيق بيانات إضافيّة للطباعة. ولأن هذا الطلب هو طلب دخل / خرج غير متزامن، يجب على شعبة التطبيق أن لا تكتب فوق محتويات المخزن المؤقّت للطباعة إلى أن تنهي الطابعة أول طلب دخل / خرج. لذلك، وقبل أن تعيد الشعبة تعبئة المخزن المؤقّت ببيانات جديدة، عليها أن تنتظر مقبض الملف الذي استعملته عند إصدار الطلب. وبعد إتمام عملية الدخل / الخرج، تضبط النواة NT مقبض الملفّ إلى الحالة المؤشّرة، وتواصل الشعبة المنتظرة التنفيذ، وربما كإعادة تعبئة المخزن المؤقّت.



الشكل (8-11)
خدمة مقاطعة جهاز (المرحلة الأولى)

8-2-2-2 خدمة مقاطعة

المرحلة الثانية لمعالجة طلب دخل / خرج موجّه إلى جهاز مُدار بمقاطعة هي خدمة مقاطعة الجهاز. فبعد أن يتمّم جهاز دخل / خرج نقل البيانات، فإنه يقاطع للخدمة وينشّط مسيّق الجهاز والنواة NT وبرنامج إدارة الدخل / الخرج. يوضح الشكل (8-11) المرحلة الأولى للمعالجة. (يتمّ شرح مسيّق الجهاز بتفصيل أكبر في القسم 8-3-1).

عند حصول مقاطعة جهاز، ينقل المعالج الذي يقبل المقاطعة التحكُّم إلى النواة، التي تفهرس في جدول توزيع المقاطعة لتحديد روتين خدمة المقاطعة (ISR) للجهاز. (يجب أن تزوّد مسيّقات الجهاز للأجهزة المدارة بمقاطعة روتين ISR، وهو روتين مسيّق يوقف مقاطعة الجهاز والمعالجات التي تطلبها المقاطعة).

مقاطعات الجهاز هي مقاطعات بأولويّة مرتفعة على معظم أنظمة التشغيل، وعادة يمنع نظام التشغيل المقاطعات بأولويّة منخفضة أو ربما كل المقاطعات إلى أن ينتهي الروتين ISR من خدمة الجهاز. لكن في النظام Windows NT، تتناول روتينات ISR مقاطعات الجهاز في خطوتين. تحصل مقاطعة الجهاز عند مستوى طلب مقاطعة مرتفع (IRQL) لكن روتين ISR يبقى عند هذا المستوى ما يكفي من الوقت لإيقاف مقاطعة الجهاز. بعد ذلك تخفّض الشعبة مستوى IRQL للمعالج وتتمّ معالجة المقاطعة. تضمن هذه الطريقة عدم منع مقاطعات البراجيّات ومقاطعات الجهاز بمستوى منخفض لأطول مما هو ضروري.

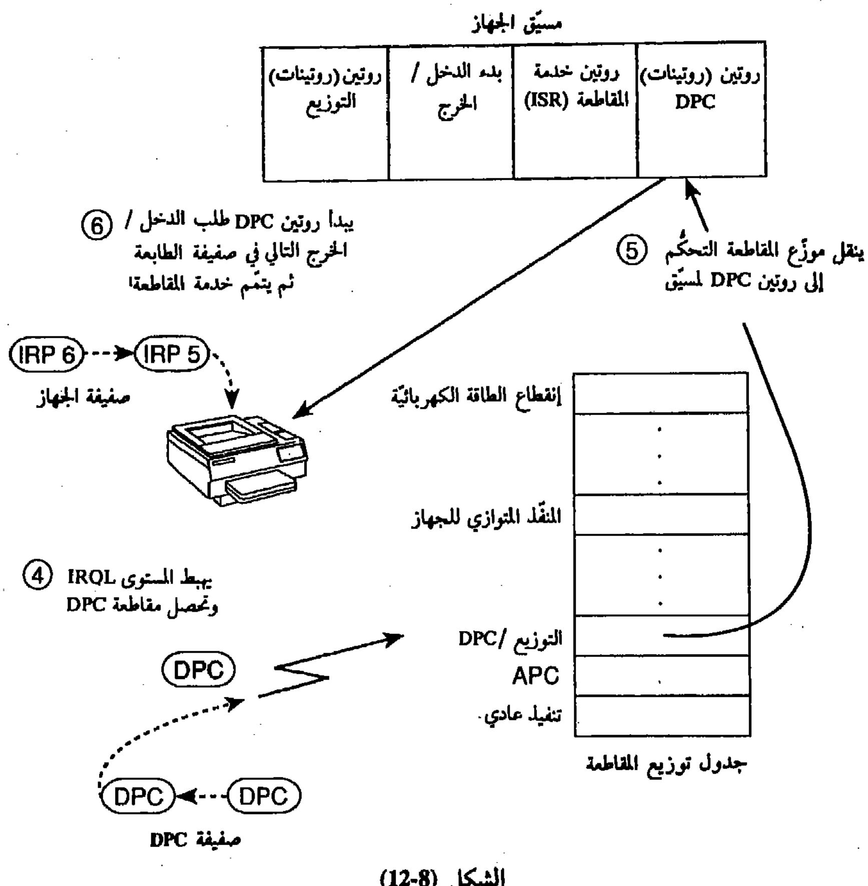
تستعمل مسيّقات الجهاز NT إستدعاءات إجراءات مؤجّلة (DPCs)، التي سبق وشُرحَت في الفصل السابع، لتنفيذ معالجة المقاطعة ذات المستويين. فمثلاً، عند حصول مقاطعة طابعة، يوقف الروتين ISR المقاطعة فوراً. وبالإعتماد على الجهاز، فإنه يستطيع القيام بذلك عن طريق قراءة مسجّل حالة جهاز. بعد ذلك يحفظ الروتين ISR أية حالة جهاز ستحتاج لها لاحقاً وتضع DPC في صفيفة ثم تنتهي. يحتوي روتين DPC بقيّة الشيفرة لمعالجة المقاطعة.

بعد إنهاء الروتين ISR، تخفّض شيفرة النواة NT مستوى IRQL للمعالج إلى المستوى الذي كان عليه قبل حصول المقاطعة. وكما ذكر في الفصل السابع، يؤدّي وضع روتين DPC في إحدى صفيفات DPC للنواة إلى حصول مقاطعة براجيّات في المرّة التالية التي يهبط فيها مستوى IRQL للمعالج دون مستوى التوزيع / DPC. (يوضح الشكل (8-12) على الصفحة التالية المرحلة الثانية لخدمة المقاطعة. (الشكل (8-12) هو تكملة للشكل (8-11)).

وكسائر المقاطعات الأخرى، تؤدّي مقاطعة DPC إلى نقل التحكُّم إلى موزّع مقاطعة

النواة. يتناول موزّع المقاطعة هذه المقاطعة عن طريق إستدعاء روتين DPC لمسيّق الجهاز. وقد يبدأ روتين DPC للطابعة، من ضمن الأشياء الأخرى، بدء طلب الدخل / الخرج التالي الذي ينتظر في صفيفة الطابعة ثم تسجيل حالة عمليّة الدخل / الخرج التي تمّت للتوّ. وبعد الإنتهاء من عمله، يستدعي DPC برنامج إدارة الدخل / الخرج لإتمام الدخل / الخرج والتخلُّص من روتين IRP.

إن فائدة إستعمال روتين DPC لتنفيذ معظم خدمات الجهاز تكمن في الإتاحة لأية مقاطعة ممنوعة بأولويّة تقع بين مستوى IRQL للجهاز ومستوى IRQL للتوزيع / DPC، من الحصول قبل حصول مقاطعة DPC بأولويّة أدنى. وهكذا تخدم المقاطعات بمستوى متوسط أكثر مما تخدم عادة.



الشكل (12-8)
خدمة مقاطعة جهاز (المرحلة 2)

3-2-2-8 إتمام طلب دخل / خرج:

بعد تنفيذ روتين DPC لمسيّق جهاز، يجب القيام بأعمال إضافيّة قبل إعتبار طلب الدخل / الخرج منتهي. تسمى هذه المرحلة الثالثة لمعالجة الدخل / الخرج، إتمام الدخل / الخرج، ويختلف ما يستلزمه وفقاً لعمليّات الدخل / الخرج المختلفة. فمثلاً، تسجل كل خدمات الدخل / الخرج نتيجة العمليّة في كتلة حالة الدخل / الخرج، وهي بنية بيانات مزوّدة من قبل المستدعي. وبشكل مشابه، فإن بعض الخدمات التي تنفّذ الدخل / الخرج المخزن مؤقّتاً تتطلّب من نظام الدخل / الخرج إرجاع البيانات إلى الشعبة المستدعية.

في كلتا الحالتين، يجب أن ينسخ نظام الدخل / الخرج بعض البيانات المخزّنة في ذاكرة النظام إلى فسحة العنوان الظاهري للمستدعي. وللوصول إلى فسحة العنوان الظاهري للمستدعي، يجب على برنامج إدارة الدخل / الخرج نقل البيانات «في سياق شعبة المستدعي»، أي خلال تنفيذ شعبة المستدعي. وهو يقوم بذلك عن طريق وضع روتين APC في نمط النواة إلى الشعبة في صفيفة، كما يبيّن في الشكل (8-13) هو تكملة للشكل (8-12).

يشبه روتين APC روتين DPC، بإستثناء تنفيذ روتين APC في سياق شعبة معيّنة، بينما يستطيع روتين DPC التنفيذ من أي سياق شعبة. إضافةً لذلك، تطلق روتينات APC وDPC مقاطعات البراجيّات لكن مقاطعات APC تحصل عند مستوى IRQL أدنى من مقاطعات DPC.

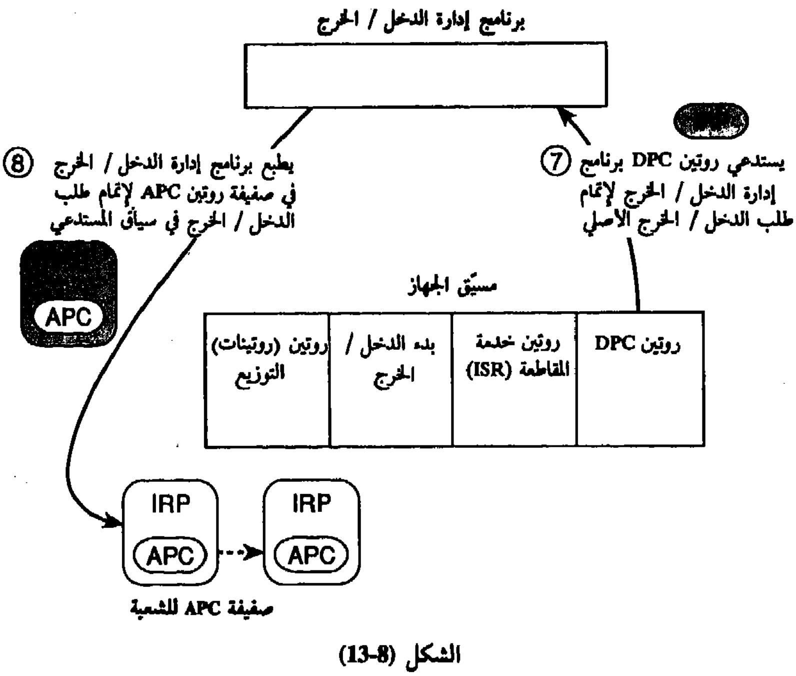
تضع شيفرة النظام (مثل برنامج إدارة الدخل / الخرج) في صفيفة روتين APC في نمط النواة إلى شعبة معيّنة عن طريق إستدعاء روتين نواة. وفي المرّة التالية عندما تبدأ الشعبة التنفيذ على مستوى IRQL المنخفض، يحصل مقاطعة براجيّات. يوضح الشكل (8-14) على الصفحة التالية المرحلة الثانية لإتمام الدخل / الخرج. (الشكل (8-14) هو تكملة للشكل (8-13)).

عند حصول مقاطعة APC، تنقل النواة التحكّم إلى روتين APC لبرنامج إدارة الدخل / الخرج، الذي ينسخ البيانات (إذا وجدت) ويرجع الحالة إلى فسحة عنوان المستدعي الأصلي ويحذف روتين IRP الذي يمثّل عمليّة الدخل / الخرج ويضبط مقبض ملفّ المستدعي (أو الحدث المزوّد من قبل المستدعي، كما يُشرح لاحقاً) إلى الحالة المؤشّرة. وبذلك يكون الدخل / الخرج قد تمّ. وتفلت شعبة المستدعي أو أية شعبة أخرى تنتظر مقبض الملفّ (المقبض إلى المنفّذ المتوازي في المثال الأصلي) من حالة الإنتظار وتعاود التشغيل.

ملاحظة أخيرة حول إتمام الدخل / الخرج: تتيح خدمات الدخل / الخرج غير المتزامنة لمستدعي تزويد روتين APC في نمط المستعمل كبارامتر. فإذا قام المستدعي بذلك، يضع برنامج

إدارة الدخل / الخرج في صفيفة روتين APC هذا إلى المستدعي على أنه الخطوة الأخيرة في إتمام الدخل / الخرج. تتيح هذه المزيّة لمستدعي تحديد عمليّة مسبقاً يريد تنفيذها عند إتمام طلب دخل / خرج. فمثلاً، قد تنفذ شعبة في نظام فرعي لمحيط عمليّة قراءة نيابة عن شعبة مستضاف. تصدر شعبة النظام الفرعي طلب دخل / خرج غير متزامن للقراءة في ملفّ وتحدّد روتين APC في نمط المستعمل. بعدها تصبح حرّة لخدمة طلبات المستضاف الأخرى.

عند إتمام الدخل / الخرج، يضع برنامج إدارة الدخل / الخرج في صفيفة روتين APC للنظام الفرعي إلى شعبة النظام الفرعي، ويقاطع المعالج الشعبة. وهكذا يحثّ النظام الفرعي لتنفيذ روتين APC الذي يرسل في هذا المثال نتائج عمليّة القراءة إلى المستضاف. بعد ذلك، تستعيد النواة سياق شعبة النظام الفرعي ويواصل النظام الفرعي من حيث توقّف قبل إستلامه مقاطعة APC. (روتينات APC في نمط المستعمل في NT مرئيّة لمبرمجي Win 32 كروتينات إتمام في روتينات API () ReadFileEx و () WriteFileEx).

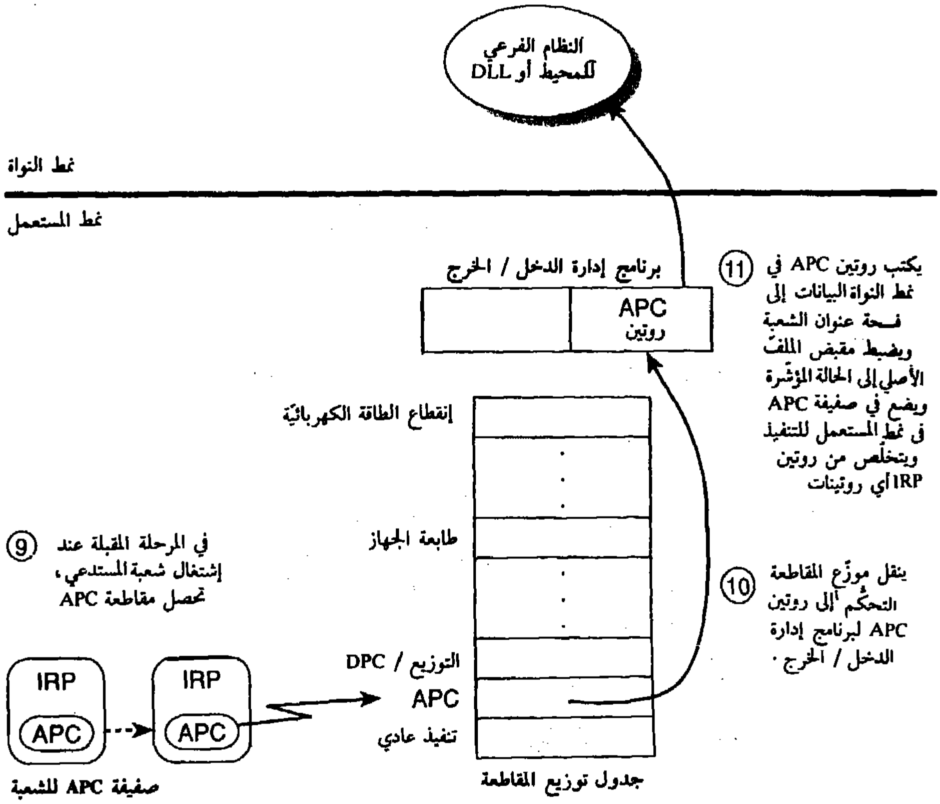


الشكل (8-13)
إتمام طلب دخل / خرج (المرحلة الأولى)

3-2-8 طلبات الدخل / الخرج إلى المسيّقات الطبقيّة:

إعتمدت الأمثلة السابقة على طلبات الدخل / الخرج إلى جهاز بسيط يتحكّم فيه مسيّق جهاز واحد. وتتشابه إلى حدٍّ كبير معالجة الدخل / الخرج لأجهزة تعتمد على ملفّ أو لطلبات إلى مسيّقات طبقية أخرى. أما الإختلاف الرئيسي فهو إضافة طبقة واحدة أو أكثر من المعالجة إلى النموذج. يظهر الشكل (8-15) على الصفحة التالية كيفيّة تحرّك طلب دخل / خرج غير متزامن عبر المسيّقات الطبقيّة. وهو يستعمل مثال قرص مضبط بواسطة نظام ملفّ.

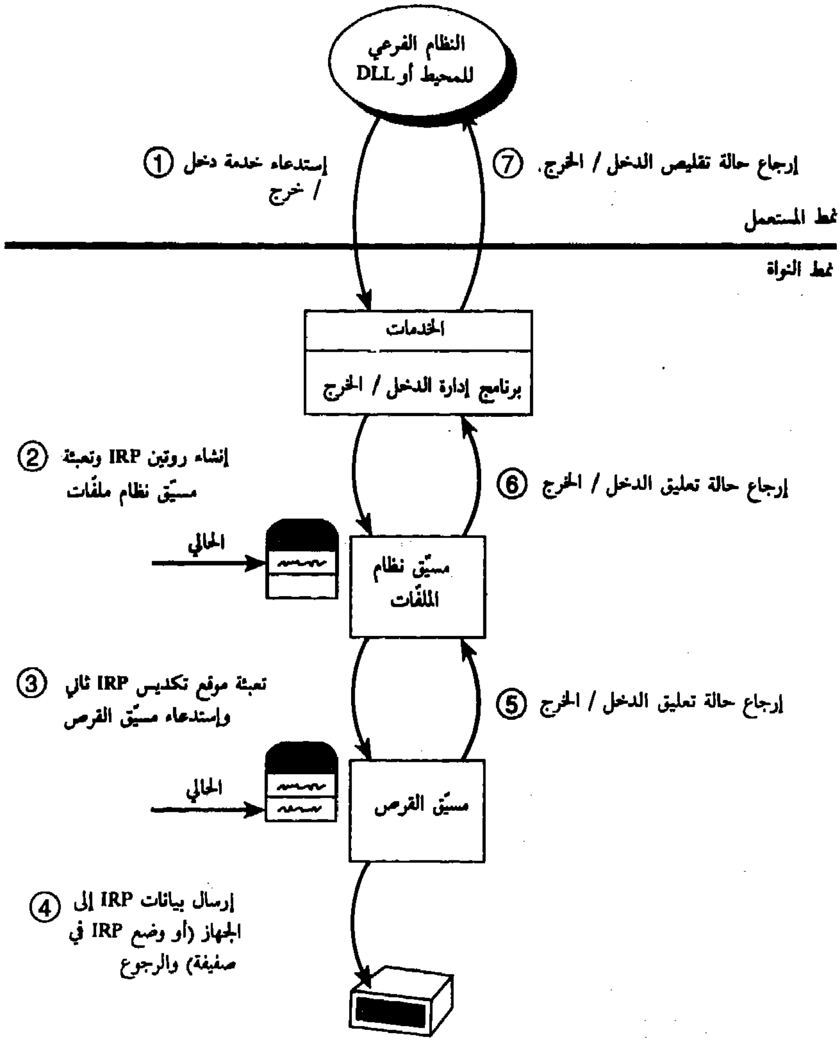
مرة أخرى، يستلم برنامج إدارة الدخل / الخرج الطلب وينشىء حزمة طلب دخل / خرج لتمثيله. لكن في هذه المرّة، فإنه يسلّم الحزمة إلى مسيّق نظام ملفّ. يتحكّم مسيّق نظام الملفّ إلى حدٍّ كبير بعمليّات الدخل / الخرج عند هذه المرحلة ووفقاً لنوع طلب المستدعي،



الشكل (8-14)
إتمام طلب الدخل / الخرج (المرحلة الثانية)

يستطيع نظام الملفّات إرسال نفس روتين IRP إلى مسيّق الجهاز، أو يمكنه إصدار حزمات طلب دخل / خرج إضافيّة وإرسالها على حِدا إلى مسيّق الجهاز.

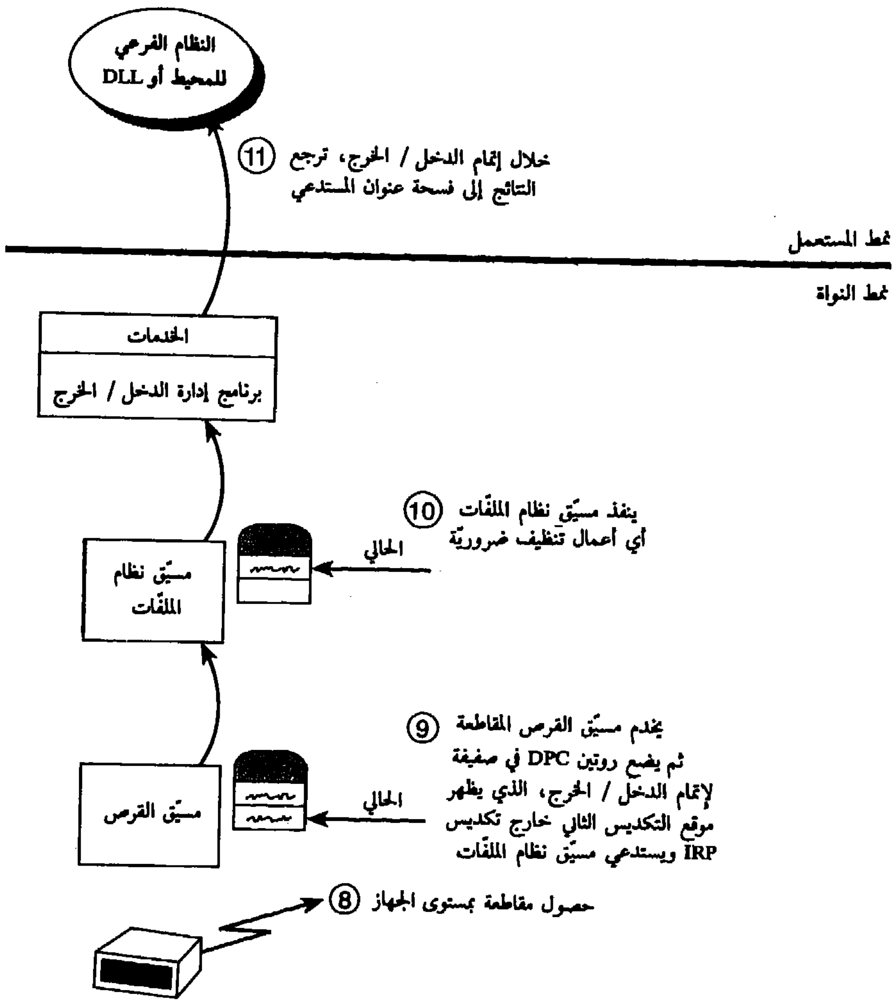
يحتمل أن يعاود نظام الملفّات إستعمال روتين IRP إذا ترجم الطلب الذي إستلمه إلى طلب واحد واضح إلى جهاز. فمثلاً، إذا أصدر تطبيق طلب قراءة لأول 512 بايت في ملفّ مخزّن على قرص مرن، يستدعي نظام الملفّات FAT مسيّق القرص ويطلب منه قراءة قطاع واحد من القرص المرن، بدءاً من موقع بدء الملفّ.



الشكل (8-15)

وضع طلب غير متزامن في صفيفة إلى المسيّقات الطبقيّة

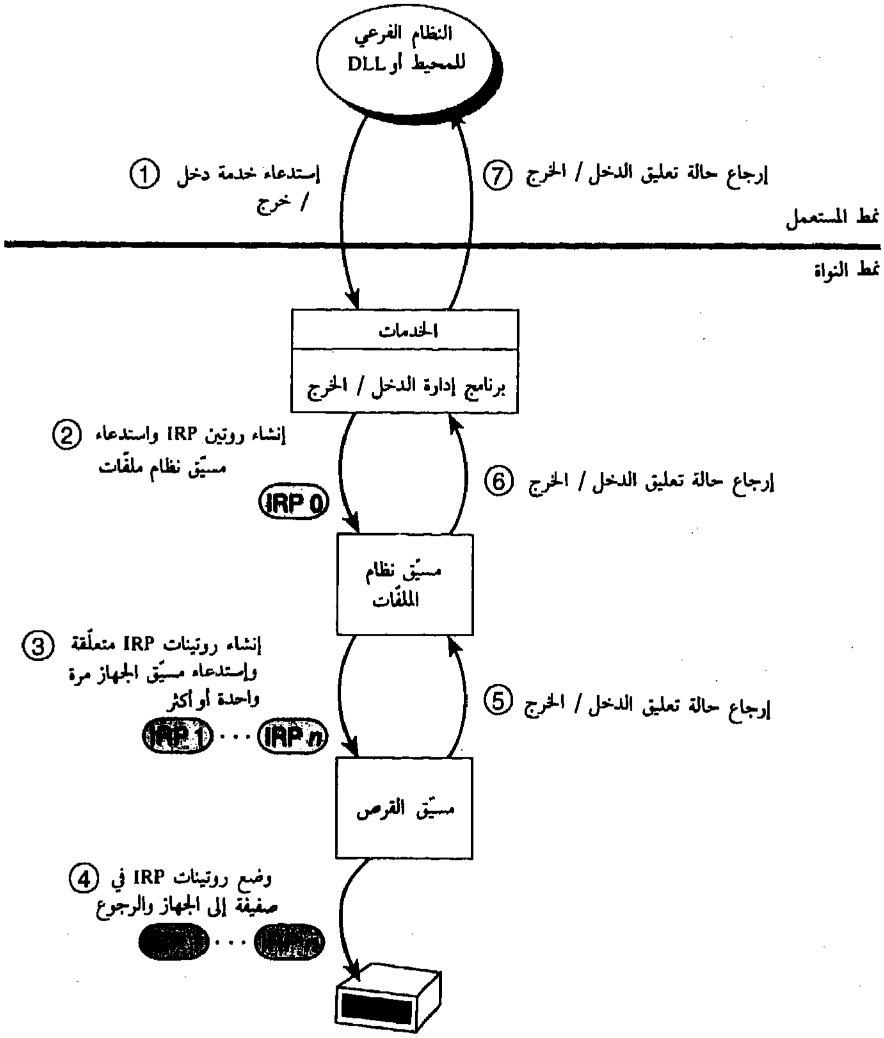
ولتكليف إعادة إستعماله من قبل عدّة مسيّقات في طلب إلى مسيّقات طبقيّة، يحتوي روتين IRP سلسلة من مواقع تكديس IRP، كما يبيّن في الشكل (8-15). تحتوي مناطق البيانات هذه، واحدة لكل مسيّق مستدعى، المعلومات التي يحتاجها كل مسيّق لتنفيذ جزء الطلب الخاص به ـ فمثلاً، شيفرة الوظيفة ـ والبارامترات ومعلومات سياق المسيّق. وكما يوضح الشكل (8-15) تعبأ مواقع التكديس الإضافيّة عندما يمرّ روتين IRP من مسيّق واحد إلى التالي. يمكن إعتبار روتين IRP مشابهاً لتكديس لجهة إضافيّة البيانات إليه وإزالتها منه خلال مدة



الشكل (8-16)
إتمام طلب دخل / خرج مرتّب في طبقات

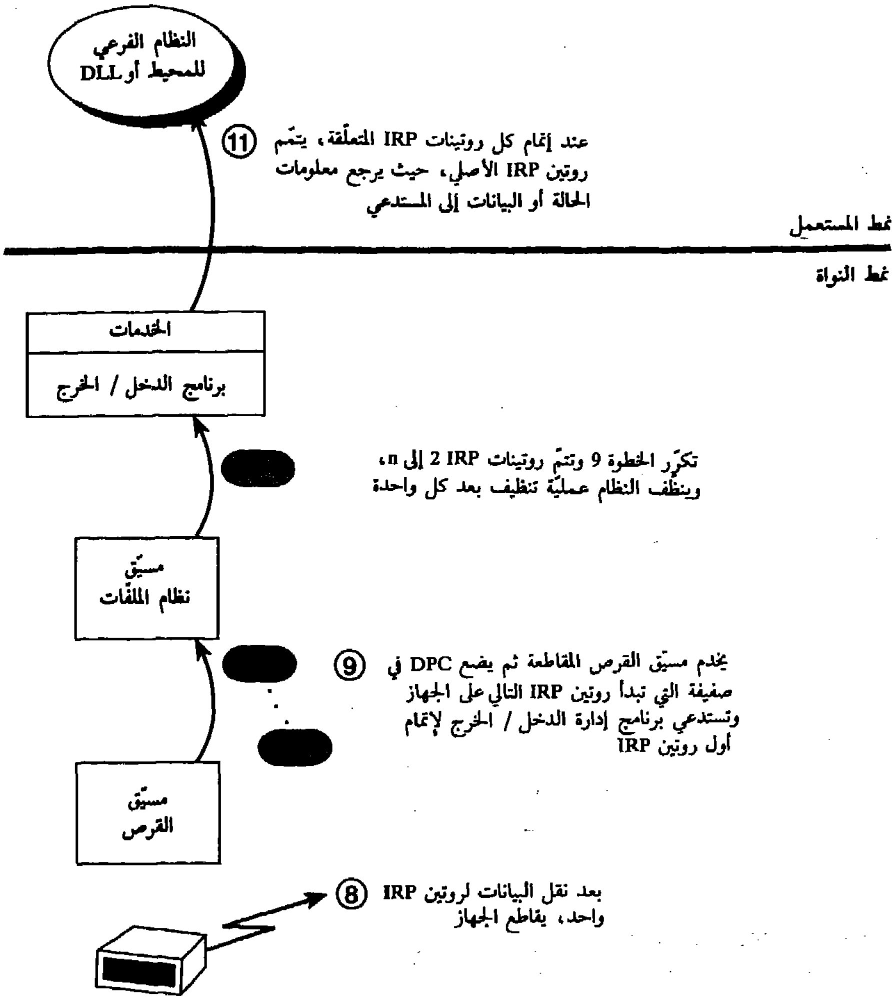
خدمته. لكن روتين IRP غير متعلّق بأية معالجة معيّنة وحجم حصّته لا يكبر أو لا يصغر. يحدّد برنامج إدارة الدخل / الخرج حصة روتين IRP من ذاكرة نظام غير مرتبة في صفحات عند بداية عمليّة الدخل / الخرج.

بعد أن ينتهي مسيّق قرص من نقل البيانات، يقاطع القرص ويتمّم الدخل / الخرج كما يبيّن في الشكل (8-16). (الشكل (8-16) هو تكملة للشكل (8-15)).



الشكل (8-17)

وضع روتينات IRP المتعلّقة في صفيفة



الشكل (18-8)
إتمام روتينات IRP المتّصلة

كطريقة بديلة لإعادة إستعمال روتين IRP واحد، يستطيع نظام ملفّ إنشاء مجموعة من روتينات IRP المتعلّقة تعمل على التوازي على طلب دخل / خرج واحد. فمثلاً، إذا كانت البيانات الواجب قراءتها من ملفّ منتشرة عبر القرص، قد ينشىء مسيّق نظام الملفّات عدّة روتينات IRP، يقرأ كل منها جزءاً من الطلب من قطاع مختلف. يوضح ذلك الشكل (17-8) على الصفحة التالية.

يسلّم مسيّق نظام الملفّات روتينات IRP المتعلّقة إلى مسيّق الجهاز الذي يضعها في صفيفة إلى الجهاز. وهي تعالج واحدة في كل مرة، حيث يتعقّب مسيّق نظام الملفّات البيانات الراجعة. وبعد إتمام كل روتينات IRP المتعلّقة، يتمّم نظام الدخل / الخرج روتين IRP الأصلي ويرجع إلى المستدعي، كما يظهر في الشكل (8-18). (الشكل (8-18) هو تابع للشكل (8-17)).

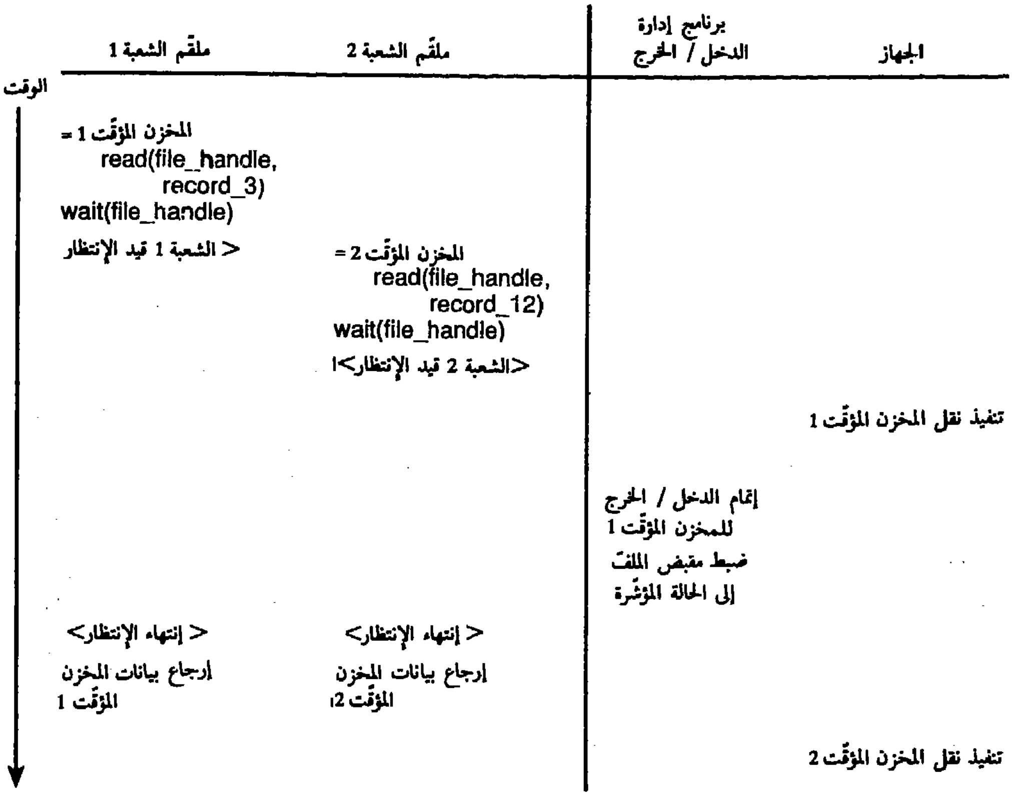
8-2-4 إعتبارات إستعمال الدخل / الخرج غير المتزامن:

عند إستدعاء خدمات دخل / خرج NT، يجب أن يختار المطوّر لجهة إستدعائها تزامنياً أو غير تزامني، وللعمليات السريعة، أو لتلك بفترة محدّدة متوقّعة، فإنه يكفي إستعمال دخل / خرج متزامن، حيث يزوّد نظام الدخل / الخرج فقط عمليّة متزامنة لهذه الخدمات. يفيد الدخل / الخرج غير المتزامن للعمليّات بأوقات طويلة جداً أو متغيّرة بكثرة. فمثلاً، يمكن أن يؤثّر عدد الملفّات المخزّنة في دليل إلى حدٍّ كبير على سرعة تحديد ملفّاته. وهكذا، تكون خدمة الإستعلام عن ملفّ دليلاً غير متزامنة إفتراضياً، كما هي قراءة الملفّ وكتابة الملفّ وإبلاغ المستدعي عندما يتغيّر الدليل والعديد من الخدمات الأخرى.

يمكن تصنيف الدخل / الخرج غير المتزامن على أنه يتطلّب برمجة أكبر مقابل تحكُّم أكبر على عمليّات الدخل / الخرج وإرتفاع كبير في الكفاية. ولا تعوّق الشعبة التي تستعمل دخل / خرج غير متزامن خلال نقل البيانات. لكن يجب على الشعبة أن تزامن إستعمالها لأي بيانات منقولة مع إتمام الجهاز لعمليّة النقل.

عند تنفيذ دخل / خرج غير متزامن، تستطيع شيفرة في نمط المستعمل إستعمال إحدى الكائنات التنفيذيّة المتعدّدة لمزامنة إستمراريّتها مع إتمام نقل الدخل / الخرج. تدعم كائنات الملفّ المزامنة، ولذلك فإن الطريقة الأسهل لمزامنة شعبة هي عن طريق إنتظار مقبض الملفّ عند نقطة ما بعد إصدار طلب دخل / خرج. وعند إتمام نقل البيانات، يضبط برنامج إدارة الدخل / الخرج مقبض الملفّ إلى الحالة المؤشّرة وتواصل الشعبة المنتظرة التنفيذ.

يوضح الشكل (8-19) المشكلة التي يمكن أن تعترض هذه الطريقة في حال توفّرَ لدى المستدعي أكثر من طلب دخل / خرج صالح واحد قيد المعالجة. إفترض أنّ ملقّم قاعدة بيانات NT المحليّة يستلم طلباً مستضافاً لقراءة سجل من قاعدة البيانات. وخلال تنفيذ هذه العمليّة، تطلب شعبة مستضاف أخرى الملقّم لقراءة سجل من قاعدة البيانات. يستعمل الملقّم، الذي فتح ملفّ قاعدة البيانات لمرّة واحدة فقط، نفس المقبض للإشارة إلى الملفّ المفتوح.



الشكل (19-8)
الإستعمال الخطأ لمقبض ملفّ

في هذا المثال، تصدّر شعبتا طلب دخل / خرج بإستعمال نفس مقبض الملفّ، ثم تنتظران المقبض. وكما يظهر الشكل (19-8)، تتمّ إحدى عمليّات الدخل / الخرج. وعندما يتمّ ذلك، يضبط برنامج إدارة الدخل / الخرج مقبض الملفّ إلى الحالة المؤشّرة التي تفلت الشعبتين من حالة الإنتظار، حيث يواصلان التنفيذ.

لسوء الحظ، ونتيجة التغييرات المشمولة في جدولة الشعَب على المعالج، ومن واقع إمكانيّة شفع شعبة من قبل شعبة أخرى في أي وقت، يتعذّر معرفة أية عمليّة من عمليّات الدخل / الخرج التي تتمّ. وفي الشكل (19-8)، يتمّ فقط نقل واحد للبيانات، لكن كلا الشعبتين ترجعان البيانات إلى مستضافيها. لذلك، يستلم أحد المستضافات بيانات خطأ.

إحدى حلول هذه المشكلة يكمن في تجنُّب المزامنة مع مقبض ملفّ عندما تصدر أكثر من شعبة واحدة طلبات بإستعمال نفس المقبض. وعِوَضاً عن ذلك، يمكن لكل شعبة أن تنتظر

كائنات حدث تنفيذي مستقلّ بإستعمال كائن واحد لكل طلب دخل / خرج. وتتيح كل خدمات الدخل / الخرج في NT غير المتزامنة للمستدعي تزويد مقبض إلى كائن حدث لهذا الغرض. وبشكلٍ بديل، يستطيع المستدعي تحديد روتين APC ينفّذ وظيفة بعد إتمام طلب دخل / خرج معيّن (وظيفة ترجع البيانات إلى المستضاف، على سبيل المثال).

يكمن حلّ آخر في تحديد تنفيذ كل خدمات الدخل / الخرج بتزامن (لضمان وجود طلب دخل / خرج واحد قيد المعالجة في كل مرة). وعندما ينفّذ برنامج إدارة الدخل / الخرج طلب دخل / خرج متزامن، فإنه يسلسل أيضاً طلبات الدخل / خرج المتعدّدة. أي، في حال طلبت شعبتي دخل / خرج بإستعمال نفس مقبض الملفّ، يضمن برنامج إدارة الدخل / الخرج تأخير الشعبة الثانية إلى أن تتمّ عمليّة الدخل / الخرج للشعبة الأولى. (يستعمل النظام الفرعي OS/2 في NT، الذي يسلسل طلبات الدخل / الخرج المتعدّدة، هذه المزيّة بكثرة).

3-8 نموذج المسيّق المرتّب في طبقات:

ركّزت الأقسام السابقة مبدئياً عمل المزايا التصميميّة لنظام الدخل / الخرج وطريقة مرور طلبت الدخل / الخرج من مكان إلى آخر خلال معالجته. يصف هذا القسم بنية المسيّقات والعلاقة بين برنامج إدارة الدخل / الخرج والمسيّقات التي يستدعيها وكيفيّة إتصال مسيّق واحد مع آخر في نموذج المسيّق المرتّب في طبقات. ينتهي القسم بشرح لأهم مزيّتين تؤثّران على المسيّقات، وعلى الأشخاص الذين طوّروها.

1-3-8 بنية مسيّق:

إن مسيّق القرص المرن ومسيّق القرص الصلب المثبّت إلى الناقل العمومي SCSI ومرقاب الرسوم التخطيطيّة ونظام الملفّات FAT وجهاز الشبكة هي أجهزة دخل / خرج مختلفة جذرياً لكنها قابلة للتبادل بشكل خاص على برنامج إدارة الدخل / الخرج. يحتوي كل مسيّق NT على المجموعة القياسيّة التالية (أو مجموعة فرعيّة) من المكوّنات:

- روتين تحفيز: ينفّذ برنامج إدارة الدخل / الخرج روتين تحفيز المسيّق عندما يحمّل المسيّق في نظام التشغيل. ينشىء الروتين كائنات النظام التي يستعملها نظام الدخل / الخرج لمعرفة المسيّق والوصول إليه.
- مجموعة روتينات توزيع: روتينات التوزيع هي الوظائف الأساسيّة التي يوفّرها مسيّق جهاز. بعض الأمثلة هي وظائف القراءة والكتابة وأية قدرات أخرى يدعمها الجهاز ونظام الملفّات

أو الشبكة. وعند إستدعائه لتنفيذ عمليّة دخل / خرج، يصدّر برنامج إدارة الدخل / الخرج روتين IRP ويستدعي مسيّقاً بواسطة إحدى روتينات توزيع المسيّق.

- روتين بدء الدخل / الخرج: يستعمل المسيّق روتين بدء الدخل / الخرج لتحفيز نقل البيانات إلى جهاز أو منه.
- روتين ISR: عندما يقاطع جهاز، ينقل موزّع مقاطعة النواة التحكُّم إلى هذا الروتين. وفي نموذج الدخل / الخرج في NT، تشتغل روتينات خدمة المقاطعة عند مستوى IRQL مرتفع، بحيث تنفذ أقل عمل ممكن لتجنُّب المنع غير الضروري للمقاطعات بمستوى أدنى. يقوم روتين ISR بوضع DPC في صفيفة، الذي يشتغل عند مستوى IRQL أدنى، لتنفيذ بقيّة معالجة المقاطعة. (لاحظ أن مسيّقات الأجهزة المدارة بمقاطعة فقط تحتوي روتينات ISR. وعلى سبيل المثال، لا يحتوي نظام الملفّات على روتين ISR).
- روتين DPC لخدمة المقاطعة: ينفّذ روتين DPC معظم الأعمال المشمولة في مناولة مقاطعة جهاز بعد تنفيذ روتين ISR. وهو ينفّذ عند مستوى IRQL أدنى من مستوى روتين ISR لتجنُّب المنع غير الضروري للمقاطعات الأخرى. يحفز روتين DPC إتمام الدخل / الخرج ويبدأ عملية الدخل / الخرج التالية المصفوفة على الجهاز.
- روتين الإتمام: يستطيع مسيّق مرتّب في طبقات إنشاء روتين إتمام يبلغه إنتهاء مسيّق بمستوى أدنى من معالجة روتين IRP. فمثلاً، يستدعي برنامج إدارة الدخل / الخرج روتين إتمام نظام الملفّات بعد أن ينتهي مسيّق جهاز من نقل البيانات إلى ملفّ أو منه. يبلغ روتين الإتمام نظام الملفّات حول نجاح العمليّة أو إخفاقها أو إلغائها وهو يتيح لنظام الملفّات تنفيذ عمليّات التنظيف.
- روتين إلغاء دخل / خرج: عند إلغاء عمليّة دخل / خرج، يحدّد المسيّق روتين إلغاء الدخل / الخرج واحداً أو أكثر. يختلف روتين الإنهاء الذي يستدعيه برنامج إدارة الدخل / الخرج وفقاً لمدى تنفيذ العمليّة عندما أُلغيت. يسجّل روتين IRP روتين إلغاء الدخل / الخرج النشط في أي وقت معيّن.
- روتين إلغاء التحميل: يُفلِت روتين إلغاء التحميل أيّ موارد نظام يستعملها المسيّق لكي يزيلها برنامج إدارة الدخل / الخرج من الذاكرة. يمكن تحميل مسيّق وإلغاء تحميله خلال إشتغال النظام.
- روتينات تسجيل الخطأ: عند حصول أخطاء غير متوقّعة (مثلاً، عند تعطُّل كتلة قرص)، تسجّل روتينات تسجيل خطأ المسيّق حصول الخطأ وتبلغ برنامج إدارة الدخل / الخرج. يكتب برنامج إدارة الدخل / الخرج هذه المعلومات إلى ملفّ سجلّ الأخطاء.

يحتوي مسيّق الجهاز في شكله الأبسط، على روتين تحفيز يحمّل المسيّق في النظام وروتين إلغاء التحميل الذي يزيله. وهو يحتوي على روتين توزيع واحد لكل عملية يدعمها (أو روتين توزيع واحد يتناول كل العمليّات). كذلك تحتوي مسيّقات الأجهزة للأجهزة المدارة بمقاطعة على روتين إختياري يبدأ عمليّة الدخل / الخرج وروتين DPC لتنفيذ معالجة المقاطعة بأولويّة أدنى. إضافة لذلك، يحتوي عادة المسيّق المرتّب في طبقات على روتين إتمام.

2-3-8 كائن المسيّق وكائن الجهاز:

عندما تفتح شعبة مقبضاً إلى كائن ملفّ، يجب أن يحدّد برنامج إدارة الدخل / الخرج من إسم كائن الملفّ المسيّق (أو المسيّقات) الواجب إستدعاءها لمعالجة الطلب. إضافة لذلك، يجب أن يتمكّن برنامج إدارة الدخل / الخرج من تحديد موقع هذه المعلومات عندما تستعمل شعبة نفس مقبض الملفّ في المرّة المقبلة. تلبّي كائنات النظام التالية هذه الحاجة:

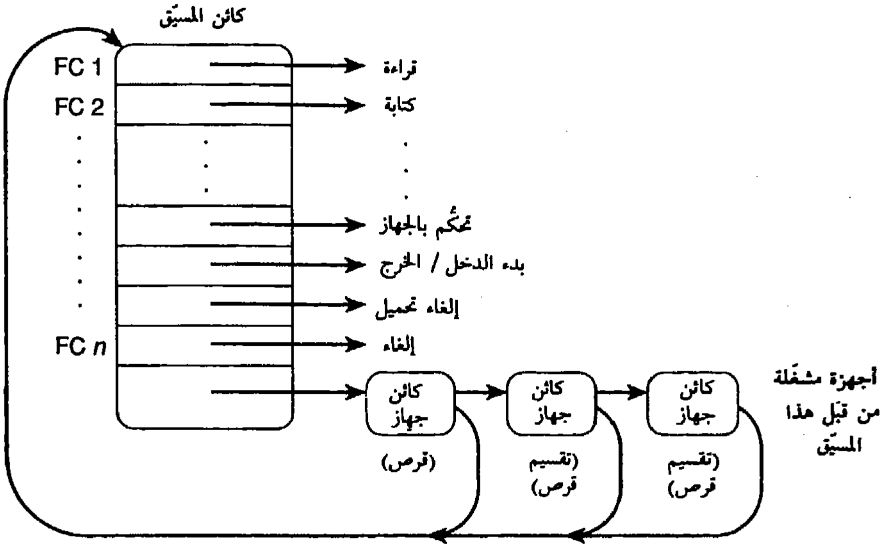
- كائن مسيّق، الذي يمثّل مسيّقاً إفرادياً في النظام ويسجّل عنوان كل من روتينات توزيع المسيّق (نقاط الإدخال) إلى برنامج إدارة الدخل / الخرج.
- كائن جهاز، الذي يمثّل جهازاً فعلياً أو منطقياً أو ظاهرياً على النظام ويصف خصائصه مثل الإستعانة المطلوبة للمخازن المؤقّتة وموقع صفيفة الجهاز للإحتفاظ بحزمات طلبات الدخل / الخرج القادمة.

ينشىء برنامج إدارة الدخل / الخرج كائن مسيّق عند تحميل المسيّق في النظام، ثم يستدعي روتين تحفيز المسيّق الذي يعبىء الكائن بنقاط إدخال المسيّق. كذلك، ينشىء روتين التحفيز كائن جهاز واحد لكل جهاز مشغّل بواسطة هذا المسيّق. وهو يزيل كائنات الجهاز عن كائن المسيّق، كما يبينّ في الشكل (8-20).

تذكر من الفصل الثالث، عندما يفتح مستدعي مقبضاً إلى كائن ملفّ، يعدّد المستدعي إسم الملفّ. يوجد ضمن هذا الإسم إسم كائن الجهاز حيث يستقرّ الجهاز. فمثلاً، يشير الإسم \ Device \ Floppy 0 \ myfile.dat إلى ملفّ يسمّى myfile.dat على سوّاقة القرص المرن A. والنضيد الفرعي \ Device \ Floppy 0 هو إسم كائن الجهاز NT الذي يمثّل سوّاقة القرص المرن. وعند فتح myfile.dat، ينشىء برنامج إدارة الدخل / الخرج كائن ملفّ ويخزّن مؤشّراً إلى كائن الجهاز Floppy0 في كائن الملفّ ثم يرجع مقبض ملفّ إلى المستدعي. بعد ذلك، وعندما يستعمل المستدعي مقبض الملفّ، يستطيع برنامج إدارة الدخل / الخرج إيجاد كائن الجهاز Floppy0 مباشرة.

كما يوضح الشكل (8-20)، يشير كائن جهاز إلى كائن مسيّق، وبهذه الطريقة يعرف برنامج

إدارة الدخل / الخرج روتين المسيّق الواجب إستدعاءه عند إستلام طلب دخل / خرج. وهو يستعمل كائن الجهاز لإيجاد كائن المسيّق الذي يمثّل المسيّق الذي يخدم الجهاز. ثم يفهرس في كائن المسيّق بإستعمال شيفرة الوظيفة المزوّدة في الطلب الأصلي. وتتوافق كل شيفرة وظيفة مع نقطة إدخال مسيّق.



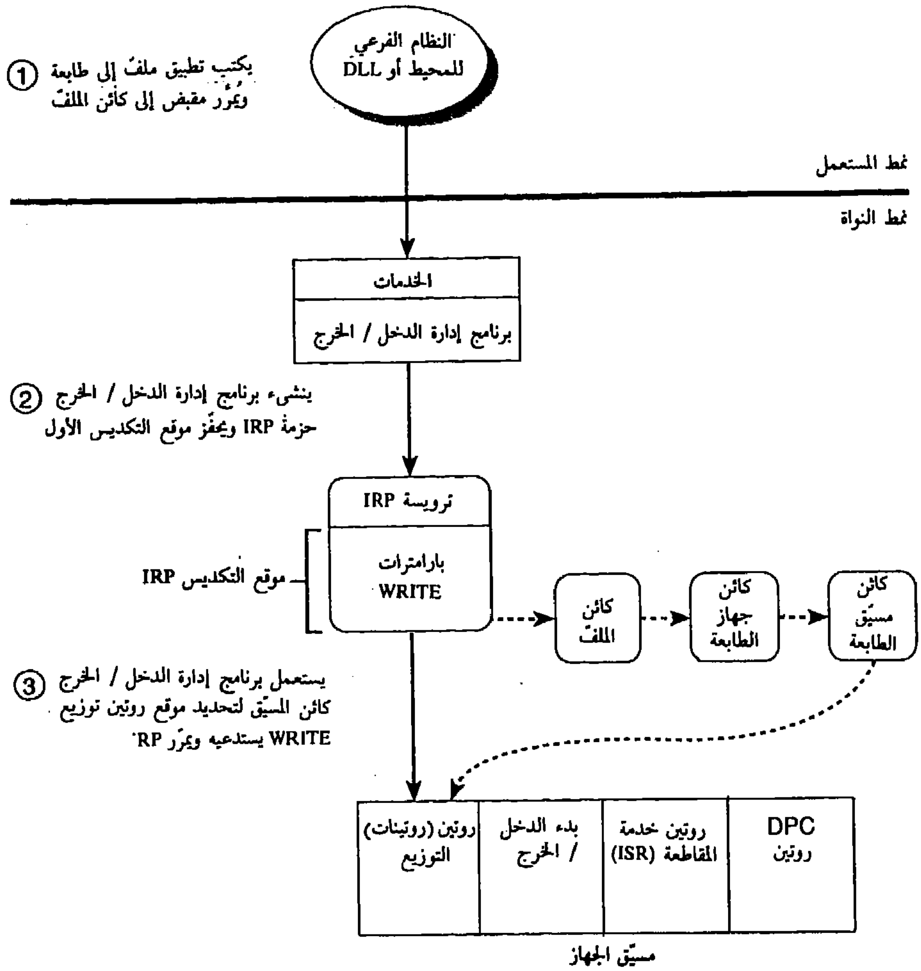
الشكل (8-20)
كائن المسيّق

يحتوي في غالب الأحيان كائن مسيّق على كائنات جهاز متعدّدة متعلّقة به. تمثّل لائحة كائنات الجهاز الأجهزة الفعليّة والمنطقيّة والظاهريّة التي يتحكّم بها المسيّق. فمثلاً، يحتوي كل تقسيم لقرص صلب على كائن جهاز مستقلّ يحتوي معلومات خاصة بالتقسيم. لكن، يستعمل نفس مسيّق القرص الصلب للوصول إلى كل التقسيمات. وعند إلغاء تحميل مسيّق من النظام، يستعمل برنامج إدارة الدخل / الخرج صفيفة كائنات الجهاز لتحديد الأجهزة التي تتأثّر نتيجة إزالة المسيّق.

يمنع إستعمال الكائنات لتسجيل المعلومات المتعلّقة بالمسيّقات برنامج إدارة الدخل / الخرج من معرفة تفاصيل المسيّقات الإفراديّة. يتبع برنامج إدارة الدخل / الخرج مؤشّر لتحديد موقع مسيّق يوفّر طبقة من النقليّة ويتيح تحميل مسيّقات جديدة بسهولة. كذلك يسهّل تمثيل الأجهزة والمسيّقات كائنات مختلفة تعيين نظام الدخل / الخرج للمسيّقات للتحكُّم بالأجهزة الإضافيّة أو المختلفة إذا تغيّر تشكيل النظام.

8-3-3 حزمة طلب الدخل / الخرج:

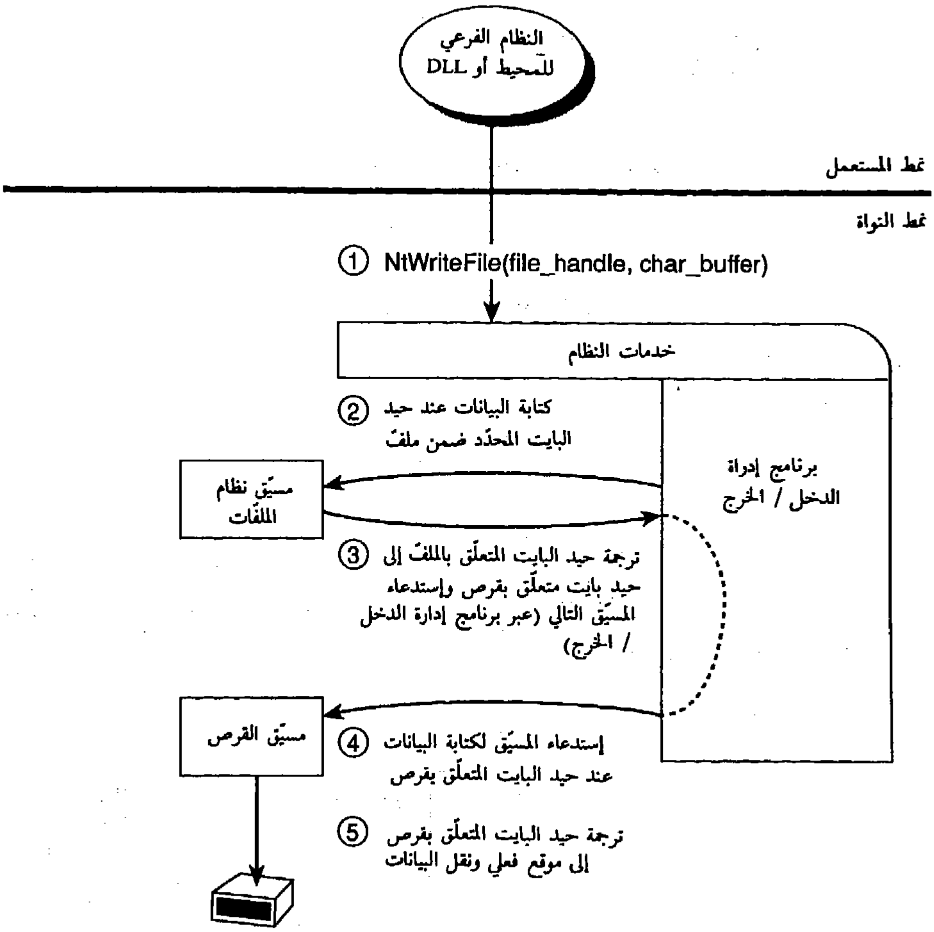
الحزمة IRP هي موقع تخزين نظام الدخل / الخرج للمعلومات المطلوبة لمعالجة طلب دخل / خرج. عندما تستدعي شعبة خدمة دخل / خرج، ينشىء برنامج إدارة الدخل / الخرج حزمة IRP لتمثّل العمليّة خلال تنفيذها عبر نظام الدخل / الخرج. يخزّن برنامج الدخل / الخرج مؤشّراً إلى كائن ملفّ المستدعي في IRP. ويظهر الشكل (8-21) على الصفحة التالية العلاقة بين IRP وكائنات نظام الدخل / الخرج.



الشكل (8-21)
إستدعاء مسيّق

تتألّف حزمة IRP من جزئين: القسم الثابت (يسمّى ترويسة) وموقع تكديس واحد أو أكثر. يحتوي القسم الثابت معلومات نوع الطلب وحجمه وإذا كان الطلب متزامناً أو غير متزامن، ومؤشّر إلى مخزن مؤقّت لدخل / خرج، مخزّن مؤقّتاً ومعلومات الحالة التي تتغيّر خلال تنفيذ الطلب. يحتوي موقع تكديس IRP شيفرة وظيفة وبارامترات خاصة بالوظيفة ومؤشّر إلى كائن ملفّ المستدعي. وفي الشكل (8-21)، يحتوي موقع تكديس IRP شيفرة وظيفة الكتابة WRITE.

خلال تنشيطها، تخزّن كل حزمة IRP في صفيفة IRP متعلّقة بالشعبة التي طلبت



الشكل (8-22)
وضع مسيّق نظام ملفّات ومسيّق قرص في طبقات

الدخل / الخرج. وهذا يتيح لنظام الدخل / الخرج إيجاد أية حزمات IRP صالحة وحذفها إذا إنتهت الشعبة أو أنهيت بطلبات دخل / خرج صالحة.

4-3-8 إضافة المسيّقات المرتّبة في طبقات:

يوضح الشكل (8-21) كيفيّة إستدعاء برنامج إدارة الدخل / الخرج لمسيّق جهاز أحادي الطبقة. يستعمل برنامج إدارة الدخل / الخرج مقبض ملفّ المستدعي لتحديد موقع الجهاز وكائنات المسيّق ثم يستدعي مسيّق الجهاز. لكن معظم عمليات الدخل / الخرج ليست كذلك. وعادة يستدعى أكثر من مسيّق واحد لمعالجة طلب دخل / خرج.

يتيح تصميم نظام الدخل / الخرج وضع مسيّق واحد على أعلى مسيّق آخر في طبقات. أي، ينفّذ مسيّق واحد فعلاً يعتمد على المعلومات المخزّنة في موقع التكديس الأول للحزمة IRP ثم يمرّر الطلب إلى مسيّق آخر، حيث يخزّن المعلومات التي يطلبها المسيّق الثاني في موقع تكديس IRP الثاني. فمثلاً، يتطلّب كتابة البيانات إلى ملفّ على قرص صلب طبقتيّ مسيّق على الأقلّ، كما يبيّن في الشكل (8-22).

يوضح هذا الشكل تقسيم العمل بين مسيّقين مرتّبين في طبقات. يستلم برنامج إدارة الدخل / الخرج طلب كتابة متعلّق ببداية ملفّ معيّن. يمرّر برنامج إدارة الدخل / الخرج الطلب إلى مسيّق نظام الملفّات، الذي يترجم عمليّة الكتابة من عمليّة متعلّقة بملفّ إلى موقع بدء (حدود مقطع على القرص) وعدد البايت للقراءة. يستدعي مسيّق نظام الملفّات برنامج إدارة الدخل / الخرج لتمرير الطلب إلى مسيّق القرص، الذي يترجم الطلب إلى موقع القرص الفعلي وينقل البيانات.

لأن كل المسيّقات ــ مسيّقات الجهاز ومسيّقات نظام الملفّات ــ تمثّل نفس إطار العمل إلى نظام التشغيل، يمكن إدراج مسيّق آخر بسهولة في التسلسل الهرمي دون تعديل المسيّقات الموجودة أو نظام الدخل / الخرج. فمثلاً، يمكن جعل عدّة أقراص تبدو وكأنها قرص واحد كبير جداً بإضافة مسيّق. يوجد مثل هذا المسيّق في النظام Windows-NT لتوفير دعم القرص السامح للأخطاء. يقع هذا المسيّق المنطقي المتعدّد الأحجام بين مسيّقَي نظام الملفّات والقرص، كما يبيّن ذلك الشكل (8-23).

لاحظ أنه في هذا المثال، فإن إضافة مسيّق آخر في التسلسل الهرمي لا يؤثّر على مسيّق نظام الملفّات أو على مسيّق القرص. وهي تنفّذ نفس النشاطات المنفّذة سابقاً. يستعمل برنامج إدارة الدخل / الخرج بنية البيانات الداخليّة التي أُنشئت عندما حمّل مسيّق وبالتالي ركّب لتحديد المسيّقات الواجب إستدعائها ولتحديد الترتيب التسلسلي للوصول إلى جهاز.

5-3-8 مسائل تطوير المسيّق:

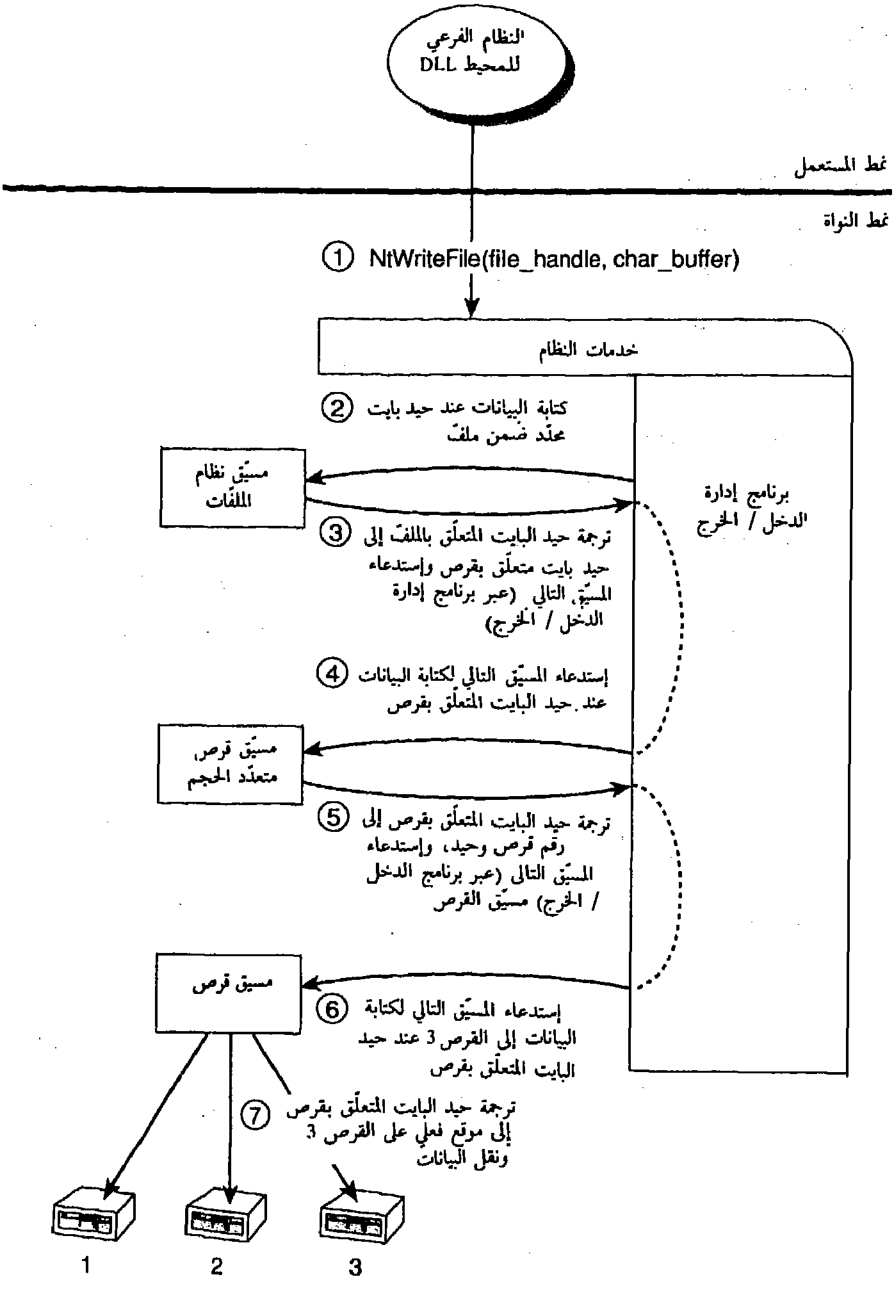
تتطلّب مزيّتان خاصتان في النظام Windows NT ـ المعالجة المتعدّدة والإستعادة بعد إنقطاع الطاقة الكهربائيّة ـ بعض الإعتبارات الإضافيّة عند إنشاء مسيّق. فمحيط المعالجة المتعدّدة هو واحد حيث يمكن أن تحصل مقاطعة على معالج واحد خلال خدمة نفس نوع المقاطعة على معالج آخر، مما يؤدّي إلى تنفيذ المسيّق على معالجتين في نفس الوقت. يدفع هذا المحيط المسيّق لإستعمال نموذج آخر لعملية بدلاً من النماذج المستعملة في أنظمة التشغيل MS-DOS و OS/2، على سبيل المثال.

كذلك، يوفّر النظام Windows NT القدرة على الإستعادة بعد حصول إنقطاع في الطاقة الكهربائيّة. ومع معظم أنظمة التشغيل، وعند إنقطاع الطاقة الكهربائيّة خلال إشتغالها، تتوقّف عادة الأنظمة، حيث تفقد كل عمليات الدخل / الخرج قيد التنفيذ. لكن من الناحية المقابلة، يجهّز النظام Windows NT بمصدر طاقة لا ينقطع (UPS)، وهو يحذّر المستعملين حول إنقطاعات الطاقة المحتومة قبل أوانها ثم توقّف النظام بطريقة نظاميّة محكومة. وإذا كان الحاسوب مزوّداً ببطاريّة مساندة للذاكرة، يوفّر النظام Windows NT قدرة إستنهاض حامية تتيح لنظام الدخل / الخرج لإستعادة عمليات الدخل / الخرج التي كانت قيد التنفيذ عند حصول إنقطاع الطاقة. يصف هذا القسم بعض الإعتبارات التصميميّة للمسيّق الناتجة عن قدرات المعالجة المتعدّدة والإستعادة من إنقطاع الطاقة في النظام NT.

1-5-3-8 المعالجة المتعدّدة:

محيط المعالجة المتعدّدة المتناظرة هو محيط تستطيع فيه نفس شيفرة النظام الإشتغال في نفس الوقت على أكثر من معالج واحد. ورغم أنه يجب على نظام التشغيل الإهتمام بالعمل بشكل صحيح على الحواسيب المتعدّدة المعالجات، فإن إعتبارات المعالجة المتعدّدة تظهر بوضوح أكبر في نظام الدخل / الخرج وخاصة في مسيّقات الجهاز.

إن الطلب الرئيسي لجعل الشيفرة تعمل في محيط متعدّد المعالجات هو حماية الموارد المشاركة لجهة عدم إستعمالها من قبَل شعبتين تنفّذان في نفس الوقت. وكسائر مكوّنات النظام الأخرى، يستطيع المسيّق التنفيذ على معالجين أو أكثر في نفس الوقت، وبالتالي يجب مزامنة وصوله إلى بيانات المسيّق بإستعمال قفل دوامي معرّف من قبَل النواة. في نظام الدخل / الخرج، تتضمّن الموارد المشاركة، إضافة إلى بنيات بيانات مسيّق الجهاز، أجهزة أحاديّة المستعمل. فمثلاً، إذا حاولت شعبتا الكتابة إلى طابعة في نفس الوقت، يجب أن لا ينثر خرجها.

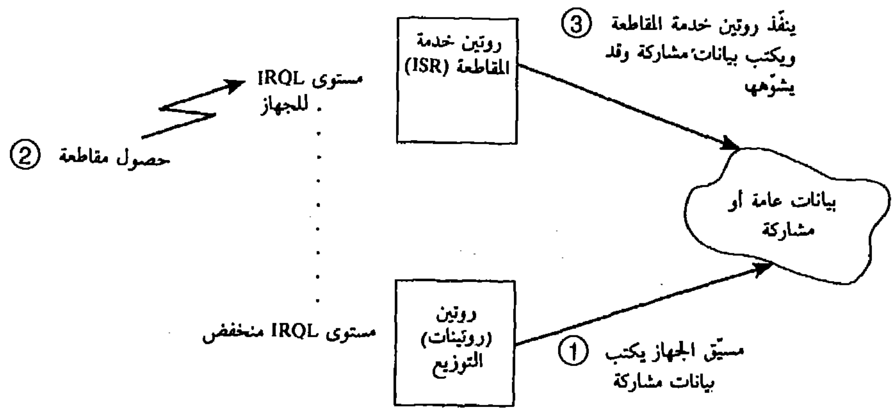


الشكل (8-23)

إضافة مسيّق مرتّب في طبقات

إن مناولة مقاطعات الجهاز بشكل صحيح على نظام متعدّد المعالجات هي مسألة مثيرة للإهتمام في نظام الدخل / الخرج. معظم شيفرة مسيّق الجهاز تشتغل عند مستوى IRQL عندما يحفز مستدعي عمليّة دخل / خرج. وللشعب في نمط المستعمل العادي، فإن هذا المستوى هو أدنى مستوى IRQL متوفّر (المستوى المنخفض أومستوى التشغيل). تحصل مقاطعات مسيّق الجهاز، من ناحية أخرى، عند مستويات IRQL المرتفعة. لذلك، يستطيع جهاز إصدار مقاطعة وتنفذ روتين ISR خلال إشتغال مسيّق الجهاز العائد له. وإذا كان مسيّق الجهاز يعدّل البيانات التي يعدّلها روتين ISR أيضاً، مثل مسجلات الجهاز والتخزين التكديسي أو البيانات الستاتيّة، قد تشوّه البيانات عند تنفيذ روتين ISR. يوضح الشكل (8-24) ذلك.

هذا السيناريو غير محدّد لأنظمة التشغيل المتعدّدة المعالجات. وقد يحصل نوع آخر من المشكلة في أنظمة التشغيل الآحاديّة المعالج. لمنع حصول المشكلة، تلغي مسيّقات الأجهزة في نظامي التشغيل OS/2 و MS-DOS على سبيل المثال، تمكين المقاطعات (بإستعمال تعليمات CLI على عتاد Intel x86)، خلال الوصول إلى مسجّلات الجهاز أو البيانات الأخرى المشاركة في روتين ISR. هذا الأمر يمنع المقاطعات ويتيح تنفيذ مسيّق الجهاز دون إمكانيّة تدخّل الروتين ISR والوصول إلى نفس البيانات.



الشكل (8-24)
تشويه البيانات في مسيّق جهاز

هذه الإستراتيجيّة فعّالة على الأنظمة الآحاديّة المعالج. وعلى الأنظمة المتعدّدة المعالج، لا يمنع إلغاء تمكين المقاطعات على معالج واحد حصول مقاطعة وخدمتها على مقاطع آخر. فمثلاً، إفترض أن شعبة تشتغل على المعالج 1 ألغت تمكين المقاطعات ثم بدأت كتابة البيانات

إلى المنفّذ المتوازي. وخلال قيام الشعبة بكتابة البيانات إلى المخزن المؤقّت للمنفّذ، تقاطع للخدمة الطابعة المثبّتة إلى المنفّذ. يستلم المعالج 2، حيث المقاطعات عليه غير ملغاة التمكين، يأخذ المقاطعة ويبدأ تنفيذ الروتين ISR للمنفّذ المتوازي. يكتب روتين ISR هذا البيانات إلى المخزن المؤقّت للمنفّذ. ويصبح المخزن المؤقّت الآن بحالة غير معروفة.

لتجنُّب هذه الحالة، يجب أن يزامن مسيّق الجهاز المكتوب للنظام NT وصوله إلى أية بيانات يشاركها مع الروتين ISR. وقبل محاولة تحديث البيانات المشاركة، يجب أن يمنع مسيّق الجهاز كل الشعَب الأخرى من تحديث نفس بنية البيانات. إضافة لذلك، يجب أن يخزّن القفل المستعمل في الذاكرة بحيث يكون عاماً لكل المعالجات. توفّر النواة NT روتينات مزامنة خاصة يجب أن تستدعيها مسيّقات الأجهزة عند وصولها إلى البيانات التي تتمكّن روتينات ISR من الوصول إليها. تمنع روتينات مزامنة النواة هذه الروتين ISR من التنفيذ خلال الوصول إلى البيانات المشاركة.

بعد هذا الشرح، يجب أن يكون قد توضّح أنه رغم حاجة روتينات ISR لإهتمام خاص، تكون أية بيانات يستعملها مسيّق الجهاز عرضة للوصول من قبل نفس مسيّق الجهاز الشغّال على معالج آخر. لذلك، يجب على شيفرة مسيّق جهاز لمزامنة إستعمالها لأيّة بيانات عامة أو مشاركة. وإذا استعملت البيانات من قبل الروتين ISR، يجب على مسيّق الجهاز إستعمال روتينات مزامنة النواة، وإلا، يستطيع مسيّق الجهاز إستعمال القفل الدوامي للنواة.

2-5-3-8 الإستعادة من إنقطاع الطاقة:

عند إنقطاع الطاقة، وحتى إذا إستردّت الطاقة بسرعة بحيث لا يلاحظ هذا الإنقطاع من قبَل الناس، تلاحظ المكوّنات الإلكترونيّة ـ وخاصة أجهزة الدخل / الخرج ـ هذا الإنقطاع. ويمكن أن تتشوّه أية بيانات مخزّنة في مسجّل جهاز، على سبيل المثال، ويمكن أن يتوقّف الجهاز نفسه أو أن يعاود ضبطه إلى حالة عشوائيّة.

ولأن مسيّقات الجهاز تنفّذ في نمط النواة مع الوصول إلى ذاكرة النظام، يمكن أن يؤدّي إنقطاع الطاقة الذي يؤثّر على إشتغال الجهاز، إلى مشاكل خطيرة في نظام التشغيل. ولمنع هذه المشاكل في النظام Windows NT، يجب أن يعرف مسيّق الجهاز متى انقطعت الطاقة، لكن لفترة مؤقّتة، ويجب أن يعيد ضبط الجهاز الذي يشغّله إلى حالة معروفة بعد الإنقطاع. كذلك، يجب معاودة بدء أية عمليّة دخل / خرج تمّ مقاطعتها، لكن في حال تعذّر ذلك، يجب إبلاغ برنامج إدارة الدخل / الخرج عن إخفاق عمليّة دخل / خرج لكي ترجع حالة خطأ إلى المستدعي.

يوفّر برنامج إدارة الدخل / الخرج، سويّة مع النواة NT، قدرة تتيح لمسيّقات الأجهزة مناولة مقاطعات إنقطاع الطاقة. فعند إنقطاع الطاقة، يحصل مقاطعة إنقطاع طاقة، حيث يحتوي نظام التشغيل على فترة زمنيّة قصيرة للتحضُّر للتوقُّف. تنسخ النواة بسرعة إلى الذاكرة كل مسجّلات النظام المهمّة بما فيها عدّاد البرنامج. وإذا زوّدت ذاكرة الحاسوب ببطاريّة دعم، تحفظ هذه المعلومات، وعندما ترجع الطاقة الكهربائيّة، تستطيع النواة ونظام الدخل / الخرج إستعمال المعلومات المحفوظة لمعاودة التنفيذ حيث توقّف. ويستطيع نظام الدخل / الخرج معاودة بدء عمليات الدخل / الخرج المقاطعة أو إنهائها.

يستطيع كل مسيّق جهاز، في NT، تنفيذ عدّة مهام لتساعده على الإستعادة من إنقطاعات الطاقة. أولاً، يستطيع كل مسيّق جهاز إنشاء وتسجيل روتين إستعادة طاقة يعيد ضبط جهازه إلى حالة معروفة بعد إنقطاع الطاقة. وبعد إستعادة الطاقة، تحدّد النواة موقع كل روتين إستعادة طاقة للمسيّق وتنفّذه.

ثانياً، يضمن كل مسيّق عدم حصول إنقطاع طاقة خلال كتابة المسيّق تتابعات من البيانات الحرجة إلى جهازه. وهذا مهمّ لأنه عند إستعادة الطاقة، يعاود تنفيذ المعالج عند عدّاد البرنامج من حيث تمّ مقاطعته، أية عملية جهاز كانت قيد التنفيذ قد تكون أتمّت أو لا بشكل صحيح، ويجب معاودة بدئها من البداية. وقبل كتابة أية بيانات حرجة إلى مسجّلات العتاد، يضمن أولاً كل مسيّق جهاز عدم حصول إنقطاع طاقة ثم يمنع مؤقّتاً مقاطعات إنقطاع الطاقة عن طريق رفع مستوى IRQL للمعالج إلى مستوى الطاقة. وبعد كتابة بياناته، يخفّض فوراً كل مسيّق جهاز مستوى IRQL (في الواقع، يستردّ مستوى IRQL السابق). وإذا حصل إنقطاع طاقة خلال ذلك، تؤجّل المقاطعة قليلاً إلى أن يتجاوز عدّاد البرنامج العمليّة الحرجة. هذا الأمر يضمن عند معاودة بدء النظام، عدم قفز مسيّق الجهاز إلى وسط عمليّة حرجة، حيث يفترض بشكل خطأ نجاح خطوة مهمّة سابقة.

4-8 بإختصار:

يتألف نظام الدخل / الخرج NT من عدد من وحدات البراجيّات، بما فيها برنامج إدارة الدخل / الخرج وسلسلة من المكوّنات التي تقع تحت تصنيف «المسيّق» الكبير. يعرّف برنامج إدارة الدخل / الخرج نموذج NT لمعالجة الدخل / الخرج وينفّذ الوظائف المشتركة مع، أو المطلوبة من قبل، أكثر من مسيّق واحد. وتكون مسؤوليّته الرئيسيّة إنشاء حزمات IRP تمثّل طلبات الدخل / الخرج ورعاية الحزمات عبر المسيّقات المتنوّعة، حيث ترجع النتائج إلى المستوى عند إتمام دخل / خرج.

تشمل المسيّقات إضافة إلى مسيّقات الأجهزة الإصطلاحيّة، نظام الملفّات والشبكة والأنبوب المسمّى والمسيّقات الأخرى. تعتمد كل المسيّقات بنية مشتركة وتتّصل مع بعضها البعض ومع برنامج إدارة الدخل / الخرج باستعمال نفس الآليات. يقبل برنامج إدارة الدخل / الخرج طلبات الدخل / الخرج ويحدّد موقع المسيّقات المتنوّعة بإستعمال كائنات نظام الدخل / الخرج، بما فيها كائنات المسيّق والجهاز. ولأن المسيّقات تمثّل بنية مشتركة لنظام التشغيل، فإنه يمكن وضعها في طبقات فوق بعضها البعض لتحقيق المنظوميّة وتخفيض الإستنساخ بين المسيّقات. داخلياً، يعمل نظام الدخل / الخرج في NT، غير تزامني لتحقيق أداء مرتفع وتوفير قدرات دخل / خرج متزامنة وغير متزامنة إلى الأنظمة الفرعيّة في نمط المستعمل.

يختلف تصميم المسيّقات للنظام NT عن تصميم المسيّقات لأنظمة التشغيل الأخرى لأنه يجب أن تعمل المسيّقات على أنظمة متعدّدة المعالجات حيث تشارك في إجراءات الإستعادة من إنقطاع الطاقة في NT. لكن، تكتب كل المسيّقات في لغة بمستوى مرتفع لتقليص وقت التطوير وتحسين نقليّتها. يعالج الفصل التالي تشبيك الحواسيب في النظام Windows NT وهو موضوع يتضمّن مسيّقي NT خاصّين: معاودة التوجيه وملقّم Windows NT.

9

تشبيك الحواسيب

حتى مؤخّراً، كانت شبكات الحاسوب الشخصي تُضاف عموماً إلى أنظمة التشغيل الموجودة عند ضرورة تحقيق الإتصال ما بين الحواسيب. فمثلاً، يسمّى Microsoft LAN Manager «نظام تشغيل الشبكة» لكنّه في الواقع مجموعة من التطبيقات المعقّدة والمسيّقات التي تضيف قدرات تشبيك إلى أنظمة التشغيل الموجودة ــ وبشكل خاص، إلى MS-DOS و OS/2 و UNIX. وهو يزوّد وسائل مثل حساب المستعمل وأمان الموارد وآليّة الإتصال بين الحواسيب بما فيها الأنابيب المسمّاة والشقوق البريديّة. رغم إستمرار النماذج السابقة لبرنامج Lan Manager بتنفيذ هذه الوظائف الحرجة لأنظمة التشغيل الأخرى، على Windows NT، لا تستخدم حالياً براجميّات تشبيك الحواسيب كطبقة مضافة إلى نظام التشغيل. لكنها جزء متكامل من البرنامج التنفيذي NT مع شمل القدرات المسرودة أعلاه في نظام التشغيل.

وهكذا، ماذا يعني تشبيك الحواسيب المركّبة في الداخل؟ توجد عدّة أجوبة لهذا السؤال، لكن الإجابة الرئيسيّة هي شمل براجميّات تشبيك الحواسيب الند إلى الند (التي تسمّى أيضاً مجموعة العمل) في نتاج النظام Windows NT. يجهّز النظام لدعم نسخ الملفّات والبريد الإلكتروني والطباعة البعيدة دون الحاجة لأن يركّب المستعمل براجميّات ملقّم خاصة على أي من الماكنات. ولأن تشبيك الحواسيب هو جزء متكامل من نظام التشغيل، تستخدم براجميّات الشبكة تداخلات نظام التشغيل الداخليّة المستعملة من قبل المكوّنات الأخرى للبرنامج التنفيذي NT لإستمثال أدائها. يشرح هذا الموضوع بتفصيل لاحقاً في هذا الفصل.

بالإضافة إلى بناء قدرات تشبيك في نظام التشغيل، فإن الهدف الأساسي لتصميم تشبيك الحواسيب في NT، كما وصف من قبل Dave Thompson، مدير تطوير فريق الشبكة NT، هو توفير لبراجميّات الشبكة ومصنّعي عتاد الشبكة القَدِرَة على التركيب والتشغيل الفوري مع NT. وبهذا فإنه كان يعني قدرة الشبكات الموجودة ومسيّقات الشبكة وملقّمات الشبكة (مثل Novell NetWare و Banyan VINES و Sun NFS) على التفاعل وتبادل البيانات مع أنظمة Windows NT

بسهولة، وبوجود العديد من بروتوكولات الشبكة والبطاقات والمسيّقات المطلوب دعمها، يجب أن تعتمد Microsoft على البائعين الآخرين للمساعدة في إنتاج أجزاء متنوّعة من براجيّات شبكة Windows NT. وقد سهّلت هذه المهمّة لأن النظام Windows NT يحتوي آليّات تمكّن تحميل براجيّات الشبكة الداخليّة إلى نظام التشغيل ومنه. يمكن إستعمال نفس الآليات لتحميل براجيّات شبكة أخرى إلى نظام التشغيل ومنه.

إضافة إلى إتاحة تحميل (وإلغاء تحميل) براجيّات الشبكة، تتضمّن أهداف تشبيك الحواسيب في Windows NT على ما يلي:

- التشغيل البيني مع النماذج الموجودة من برنامج LAN Manager الشغّالة على أنظمة التشغيل الأخرى.
- الإتاحة للتطبيقات الوصول إلى أنظمة ملفّات غير عائدة إلى Microsoft على شبكات غير LAN Manager دون تعديل شيفرتها.
- توفير وسائل مناسبة لإنشاء التطبيقات الموزّعة و«You bet your business» مثل ملقّم Microsoft SQL وتطبيقات معالجة المعاملات، وما شابه.

يقدّم هذا الفصل بعض المزايا التي تجعل من تشبيك الحواسيب في Windows NT عمليّة مميّزة. يصف القسم الأول مكوّنات التشبيك الرئيسيّة وتوصيلاتها إلى منتجات تشبيك الحواسيب السابقة من Microsoft. يسهب القسم الثاني في شرح معنى «تشبيك الحواسيب الداخلي». يشرح القسم الثالث تصميم الشبكة المفتوحة في Windows NT الذي يتيح تحميل براجيّات LAN Manager و NetWare ومكوّنات الشبكة الأخرى إلى نظام التشغيل ديناميا. يصف القسم الرابع بعض الطرق حيث يجهّز Windows NT لدعم التطبيقات الموزّعة عبر أنابيبه المسمّاة والشقوق البريديّة ووسائل إستدعاء الإجراءات عن بُعد (RPC). يصف القسم الأخير وسائل تشبيك الحواسيب المتقدّمة وتسهيلات أمان الموزّع في Windows NT، التي تدعم حاجات شبكات حواسيب كل الشركة الكبيرة.

1-9 الخلفيّة:

تشبيك الحواسيب موضوع معقّد يعود إلى ماضٍ بعيد ــ رغم أن تاريخ تشبيك الحواسيب يعود إلى عقدين فقط. ففي مراحله الأولى، كان التشبيك يعني توصيل حاسوبين بواسطة سلك وإتاحة نقل الملفّات من حاسوب إلى آخر عبر السلك. وعلى مرّ الزمن، طوّر مصنّعو الحواسيب تصميم شبكة مزيد عمل ضمن أنظمتهم لكنه لم يعمل عبر الأنواع الأخرى من الأنظمة. لكن

حالياً، من الطبيعي أن يمتلك الأفراد أو الشركات شبكة عتاد حواسيب، التي يجب أن تتّصل مع بعضها.

لكن تظهر المشكلة في بنيات الشبكة المختلفة ومن قبل أجهزة الدخل / الخرج التي يجب أن تدعمها أنظمة التشغيل. فعدم التوافقيّة تسود، ويجب إنشاء نموذج تتلاءم فيه كل المكوّنات المختلفة. وفي Windows NT، تستخدم بكثرة براجيّات التشبيك كسلسلة من التمديدات المعقّدة إلى نظام الدخل / الخرج NT. وهذا الأمر منطقي إذا إعتبرت التشبيك كوسيلة يستطيع عبرها المستعملون والتطبيقات الوصول إلى الموارد البعيدة إضافة للموارد المحليّة، مثل الملفّات والأجهزة وبالتالي المعالجات.

وقبل شرح براجيّات تشبيك الحواسيب في Windows NT، يشرح أول قسمين مشيرات بعض مكوّنات التشبيك في Windows NT ثم يشرح كيفيّة ملاءمة هذه المكوّنات في النموذج القياسي لتشبيك الحواسيب.

1-1-9 التاريخ 2:

يعود تاريخ Microsoft في تشبيك الحواسيب إلى الإصدار 3.1 للنظام MS-DOS. لقد أضاف إصدار MS-DOS هذا إلى نظام الملفّات FAT لواحق قفل الملفّات وقفل السجلات الضروريّة التي أتاحت لأكثر من مستعمل واحد الوصول إلى ملفّات MS-DOS. ومع إطلاق الإصدار 1<3 للنظام MS-DOS في العام 1984، أطلقت Microsoft أيضاً إنتاجاً سُمّي Microsoft Networks والذي سُمّي عموماً «MS-NET».

حقّق MS-NET بعض الإصطلاحات التي اعتمدت لاحقاً في LAN Manager وفي Windows NT حالياً. فمثلاً، عند إصدار مستعمل أو تطبيق طلب دخل / خرج لملفّ عن بعد أو دليل أو طابعة، يكشف MS-DOS مرجع الشبكة ويمرّره إلى مكوّن براجيّات MS-NET يسمّى مغيّر الوجهة. يقبل مغيّر الوجهة في MS-NET الطلب ويرسله أو يغيّر وجهته إلى ملقّم عن بُعد. لكن رغم إعادة تصميمه بالكامل وبعد أن أصبح أكثر تعقيداً، يتضمّن تشبيك الحواسيب في Windows NT أيضاً مغيّر الوجهة. وفي الواقع، يستطيع شمل عدّة مغيّرات وجهة كل منها يوجّه طلبات الدخل / الخرج إلى أجهزة أو أنظمة ملفّات عن بُعد.

إحدى أجزاء MS-NET الأخرى التي أتبعت في Windows NT هي بروتوكول منع رسالة الملقّم (SMB). والبروتوكول هو مجموعة من القواعد والمصطلحات التي هي وحدتين مستقلّتين ــ وفي هذه الحالة، الإتصال بين الحواسيب. تتألّف عموماً براجيّات تشبيك الحواسيب من

مستويات متعدّدة من البروتوكولات مرتّبة في طبقات فوق بعضها البعض. ووفقاً للحواسيب التي يتخاطب معها النظام، فإنها قد تدعم (كما يفعل النظام Windows NT) عدّة بروتوكولات مختلفة عند مستويات مختلفة في التسلسل الهرمي. إن بروتوكول SMB في MS-NET هي مواصفات عالية المستوى لتنسيق الرسائل الواجب إرسالها عبر الشبكة. واستُعمل روتين API الذي يسمى نظام التداخل NetBIOS لتمرير طلبات الدخل / الخرج المبنيّة في نسق SMB إلى حاسوب عن بعد. ولقد إعتمد بروتوكول SMB و NetBIOS API في العديد من تشبيكات الحواسيب الصناعيّة وظهر في النظام Windows NT أيضاً.

البند الأخير الذي اعتمد من MS-NET هو ملقّم الشبكة. لقد كان ملقّم شبكة MS-NET عبارة عن براجيّات مستقرّة على حاسوب عن بعد تحوّلها إلى ملفّ مخصّص وملقّم طباعة. وهذه البراجيّات تراقب توصيل الشبكة وتنتظر وصول رسائل SMB. ومن ثم، تقوم بإلغاء توضيحها وتحدّد ما تطلبه حيث تنفذ العملية (كقراءة بيانات من ملفّ) ثم ترسل النتائج بواسطة رسالة SMB أخرى. وقد استعمل التعبير ملقّم في غالب الأحيان في سياق تشبيك الحواسيب لتعينّ حاسوباً ثم إعداده لقبول الطلبات من حاسوب عن بعد. لكن، يمكن إعتبار ملقّم الشبكة معادلاً وظيفياً لملقّم محلي (نظام فرعي محمي في مصطلحات Windows NT) يقبل الطلبات من معالجة على ماكنة أخرى عوَضاً من معالجة على نفس الماكنة.

تتضمّن براجيّات تشبيك الحواسيب في Windows NT ملقّم شبكة ند إلى ند أساسي يستعمل بروتوكول SMB (بحيث يجعله متوافقاً مع MS-NET و LAN Manager). إضافة لذلك، يستطيع Windows NT تحميل ملقّمات الشبكة الأخرى وتشغيلها إلى جانب الملقّم الداخلي. وبالنسبة للمؤسّسات المتطوّرة أو المزوّدة بشبكات كبيرة، يمكن توفير بند إضافي يسمى LAN Manager للنظام Windows NT. وهو سيحوّل محطات العمل المشبّكة نداً إلى ندّ إلى ملقّم حقل متقدّم. يستطيع ملقّم حقل مشاركة حسابات المستعمل ومعلومات الأمان مع الأنظمة المتعلّقة المتعدّدة المجموعة سويّة في حقل شبكة ومع حقول الشبكة الموثوقة الأخرى. وهو يزوّد الوسائل لتمكين الأقراص السامحة للأخطاء والمزايا المتقدّمة الأخرى. تتيح هذه القدرات للنظام Windows NT دعم متطلّبات شبكات الشركات الكبيرة.

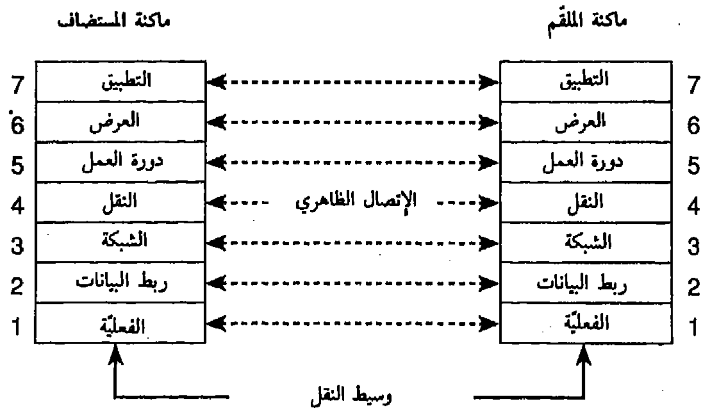
كذلك تتضمّن MS-NET مجموعة من البرامج الخدماتيّة وتركيب نحوي لأمر للوصول إلى الأقراص والطابعات عن بعد. وكما يتوقّع، فهي تتضمّن التسمية NET USE X:\\H SERVER \ SHARE. تشير الأسماء الملحقة بالنضيد \\ إلى أسماء الموارد على الشبكة التي تسمّى مصطلح التسمية المتناسق (UNC).

2-1-9 النموذج المرجعي OSI:

عرّف Andrew Tanenbaum في كتابه Computer Networks، شبكة الحاسوب على أنها «مجموعة موصولة بينياً من الحواسيب المستقلّة ذاتياً». أي، كل حاسوب مستقلّ فعلياً ويشغّل نظام التشغيل الخاص به. وهذا هو محيط تصميم شبكة Windows NT.

إن هدف براجيّات الشبكة هو تلقّي طلب (عادة طلب دخل / خرج) من تطبيق على ماكنة واحدة وتمريره إلى ماكنة أخرى وتنفيذ الطلب على ماكنة عن بُعد وإرجاع النتائج إلى الماكنة الأولى. يتطلّب تحقيق ذلك تحويل الطلب عدّة مرّات خلال تمريره. يتطلّب طلب بمستوى مرتفع مثل «قراءة عدد x من البايت من الملف y على الماكنة z»، براجيّات لتحديد كيفيّة الوصول إلى الماكنة z وبراجيّات الإتصال التي تفهمها تلك الماكنة. بعد ذلك، يجب تعديل الطلب للإرسال عبر شبكة ــ مثلاً، مقسومة إلى حزمتين قصيرتين من المعلومات. وعندما يبلغ الطلب الجهة الأخرى، يجب التدقيق فيه لجهة كماليّته وتفكيك شيفرته وإرساله إلى مكوّن نظام التشغيل الصحيح لتنفيذ الطلب أخيراً، يجب تشفير الجواب لإرساله عبر الشبكة.

لمساعدة مصنّعي الحواسيب المتعدّدين على جعل براجيّات تشبيك الحواسيب قياسيّة ومتكاملة، عرّفت منظّمة المواصفات القياسيّة الدوليّة (ISD) نموذج براجيّات لإرسال الرسائل بين الماكنات. وهو ما يُعرَف بإسم النموذج المرجعي للتوصيل البيني للأنظمة المنفتحة (OSI). يعرّف النموذج سبع طبقات من البراجيّات، كما يبيّن في الشكل (9-1).



الشكل (9-1)
النموذج المرجعي OSI

النموذج المرجعي OSI هو مخطّط مثالي تستخدمه بعض الأنظمة بدقّة، لكنّه يستعمل في غالب الأحيان لشرح مبادىء تشبيك الحواسيب. تفترض كل طبقة على ماكنة واحدة أنها تتكلّم مع نفس الطبقة على الماكنة الأخرى. تتكلّم الماكنتان نفس اللغة أو البروتوكول عند نفس المستوى. لكن، في الواقع، يجب أن يمرّر إرسال شبكة نزولاً عبر كل طبقة على ماكنة المستضاف. وأن يرسل عبر الشبكة ثم يمرّر صعوداً عبر الطبقات إلى الماكنة المقصد إلى أن يبلغ طبقة تستطيع إستيعاب الطلب واستخدامه.

إن الغرض من كل طبقة في النموذج هو توفير الخدمات إلى طبقات أعلى واستنتاج كيفيّة إستخدام الخدمات عند الطبقات الأدنى. إن تفصيل الغرض من كل طبقة يتجاوز مجال هذا الكتاب، لكن إليك بعض الوصف الموجز:

- طبقة التطبيق: تتناول نقل المعلومات بين تطبيقي شبكة، بما فيها الوظائف مثل تدقيقات الأمن وتعريف الماكنات المشاركة وتحفيز تبادل البيانات.
- طبقة العرض: تتناول تنسيق البيانات بما في ذلك الأمور كإنتهاء الأسطر بإرجاع الحاملة / تغذية سطر (CR/LF) أو إرجاع الحاملة فقط (CR)، وإذا كانت البيانات مضغوطة أو مشفّرة، وما شابه.
- طبقة دورة العمل: تدير التوصيل بين التطبيقات المتعاونة بما في ذلك المزامنة عند مستوى مرتفع والمراقبة لتحديد التطبيق «المخاطب» والتطبيق «المستمع».
- طبقة النقل: تقسّم الرسائل إلى حزمات ويعيّن لها أرقام تسلسليّة لضمان إستلامها بالترتيب الصحيح. وهي تحمي أيضاً طبقة دورة العمل من تأثير التغييرات في العتاد.
- طبقة الشبكة: تتناول الممرّات والتحكُّم بالزحمة والتشبيك الداخلي. وهي أعلى طبقة تستوعب التركيب البنيوي للشبكة، أي، التشكيل الفعلي للماكنات في الشبكة، ونوع الكابلات المستعملة لربطها سويّة وأي تحديدات في عرض النطاق وطول الكابلات المستعملة وما شابه.
- طبقة ربط البيانات: ترسل أُطر بيانات بمستوى منخفض وتنتظر إشعار إستلامها وتُعيد إرسال الأُطر التي فقدت على الخطوط غير الجيّدة.
- الطبقة الفعليّة: تمرّر البتات إلى كابل الشبكة أو وسيط إرسال فعلي آخر.

تمثّل الخطوط المتقطّعة في الشكل (9-1) البروتوكولات المستعملة في إرسال طلب إلى ماكنة عن بُعد. وكما ذكر سابقاً، تفترض كل طبقة في التسلسل الهرمي أنها تخاطب نفس الطبقة على ماكنة أخرى وتستعمل نفس البروتوكول. إن تجميع البروتوكولات التي تمرّر الطلب نزولاً ثم صعوداً عبر طبقات الشبكة تسمى تكديس بروتوكول.

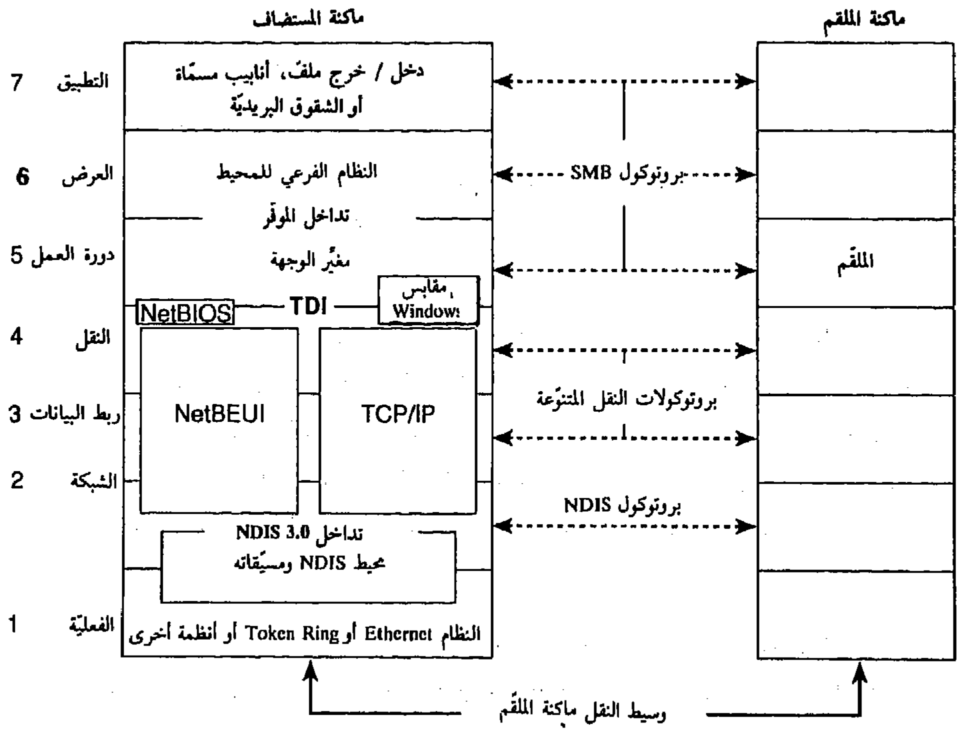
يعرض الشكل (2-9) على الصفحة التالية مكوّنات تشبيك الحواسيب في Windows NT وكيفيّة ملاءمتها في النموذج المرجعي OSI والطبقات المستعملة بين الطبقات. ويتمُّ لاحقاً وصف المكوّنات المتنوّعة في هذا الفصل.

كما يظهر الشكل، لا تتوافق طبقات OSI مع البراجيّات الفعليّة. فمثلاً، تقطع براجيّات النقل بإستمرار عدّة حدود. وفي الواقع، تسمى الطبقات الأربع السفليّة في البراجيّات جماعياً بإسم «النقل»، وتسمى مكوّنات البراجيّات التي تستقرّ في الطبقات الثلاث العليا «مستعملي النقل».

يشرح بقيّة هذا الفصل مكوّنات تشبيك الحواسيب المبيّنة في الشكل (2-9) (كذلك الأخرى غير المبيّنة في الشكل) وكيفيّة ملاءمتها سويّة وعلاقتها مع النظام Windows NT ككل.

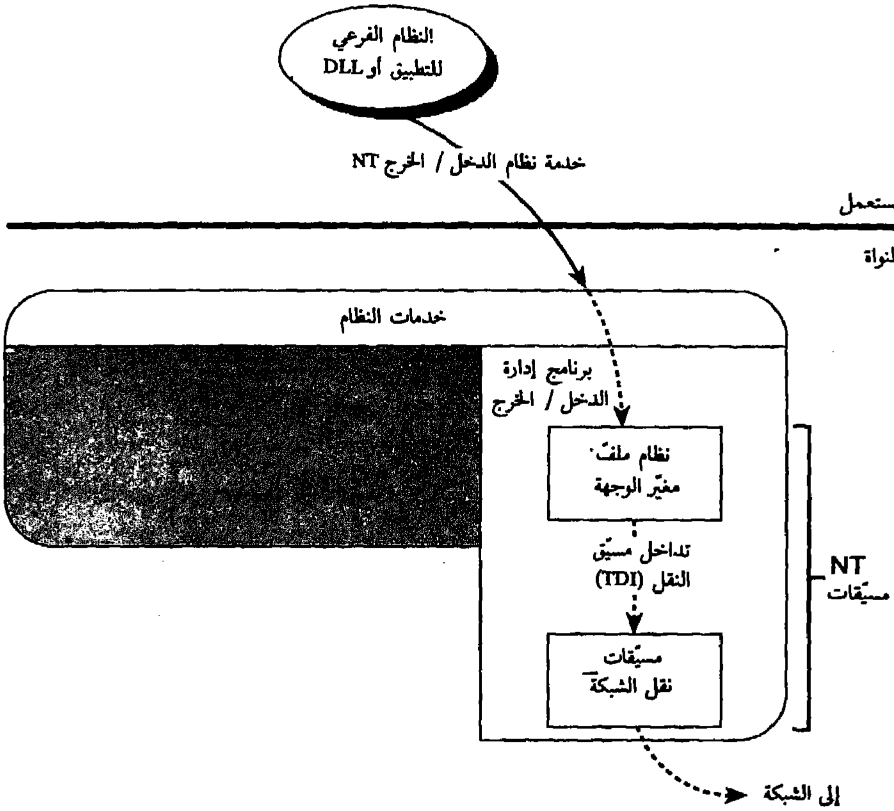
2-9 تشبيك الحواسيب الداخليّة:

أظهر القسم السابق بعض مكوّنات تشبيك الحواسيب في Windows NT الموافقة للنموذج المرجعي OSI. يظهر الشكل (3-9) كيفيّة ملاءمتها في Windows NT.



الشكل (2-9)

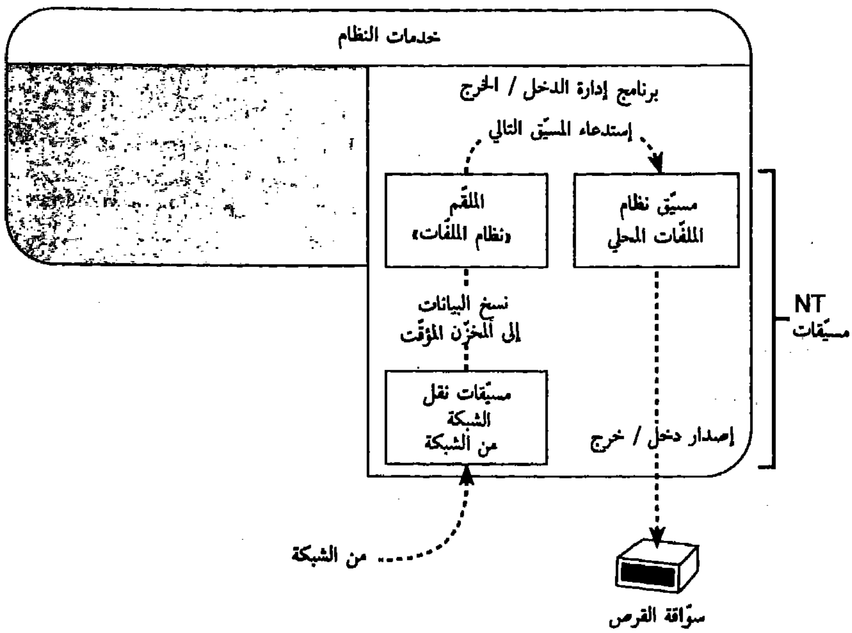
مكوّنات تشبيك الحواسيب للنموذج OSI و Windows NT



الشكل (9-3)
معاينة شبكة الدخل / الخرج المبسّطة من جهة المستضاف

تصدر برامجيّات نمط المستعمل (مثلاً، API للدخل / الخرج في Win 32) طلب دخل / خرج عن بعد عن طريق إستدعاء خدمات الدخل / الخرج المحليّة في NT. وبعد معالجة أوليّة (تُشرح لاحقاً)، ينشيء برنامج إدارة الدخل / الخرج حزمة طلب دخل / خرج (IRP) ويمرّر الطلب إلى إحدى مسيّقات نظام الملفّات المسجّل، وفي هذه الحالة مغيّر الوجهة في النظام Windows NT . يقدّم مغيّر الوجهة IRP إلى مسيّقات بطبقة أدنى (مسيّقات النقل)، التي تعالجها وتضعها على الشبكة.

عندما يصل الطلب إلى مقصد Windows NT، فإنه يستلم من قبل مسيّقات النقل ثم يمرّ عبر عدّة مسيّقات أخرى. يوضح الشكل (9-4) على الصفحة التالية إستلام طلب كتابة شبكة. يمكن لعمليّة القراءة إتباع نفس المسار إلى الملقّم، مع إرجاع البيانات إلى المسار العكسي. ويعرض لاحقاً في هذا الفصل شرح مفصّل حول معيد التوجيه والملقّم ومسيّقات النقل.



الشكل (4-9)
معاينة دخل / خرج الشبكة من جهة الملقّم

1-2-9 روتينات API للشبكة:

إعتبر كيف يمكن للتطبيق من الوصول إلى الشبكة. يزوّد النظام Windows NT عدّة وسائل:

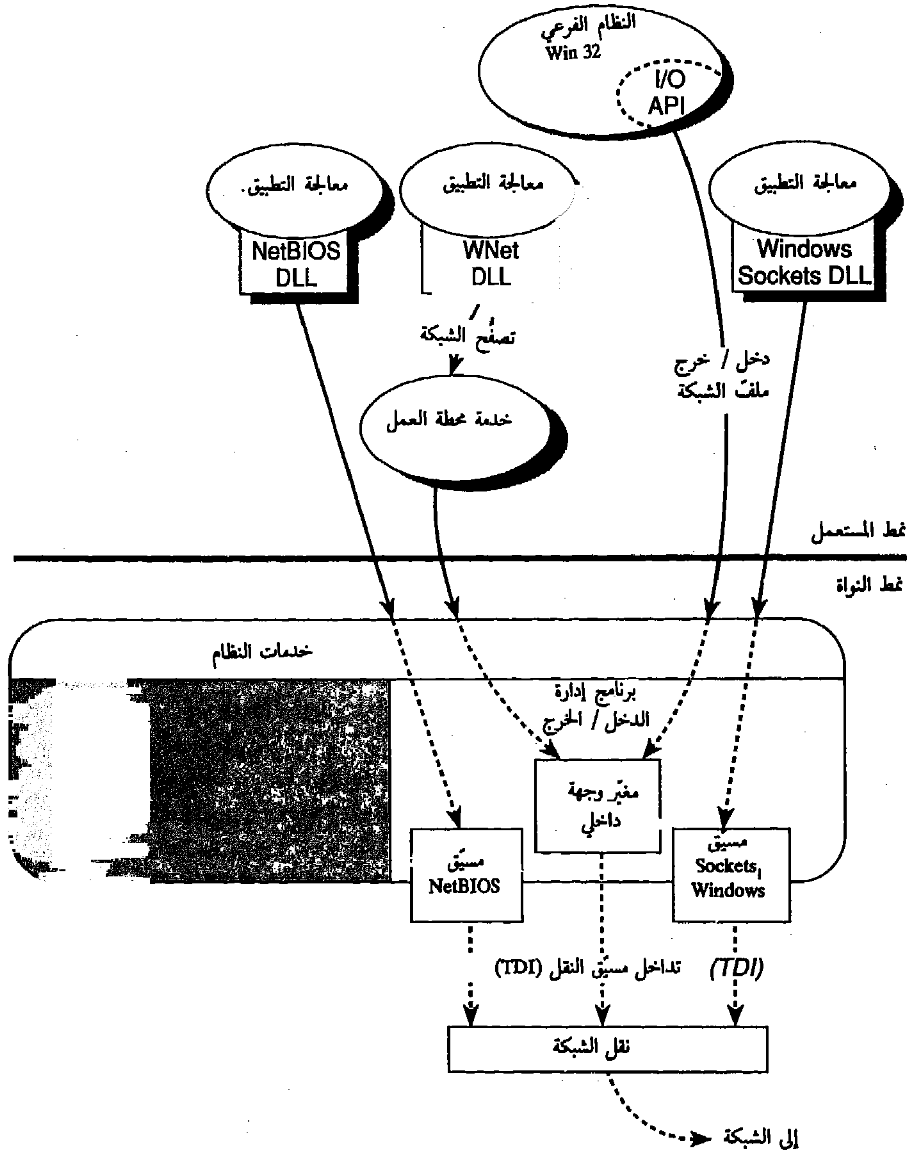
- روتين API للدخل / الخرج في Win 32. ينفّذ روتين الدخل / الخرج وظيفة الفتح والإغلاق والقراءة والكتابة العاديّة ووظائف أخرى. وهي تمرُّ عبر الشبكة فقط عندما يخرج الملفّ أو الأنبوب المسمّى الواجب الوصول إليه على ماكنة عن بُعد. وعموماً فإن ذلك يعني أن إسم الملفّ هو إسم UNC أو أن الإسم يبدأ بحرف سوّاقة يشير إلى ماكنة عن بُعد.
- روتين API للشبكة (WNet) في Win 32. هذه الروتينات مفيدة للتطبيقات مثل Windows File Manager (برنامج إدارة الملفّات في Windows) الذي يتيح التوصيل إلى أنظمة الملفّات عن بُعد وتصفّحها. ويمكن إستعمال روتينات WNet لتصفّح أنظمة ملفّات Microsoft أو الأخرى على شبكة LAN Manager, NetWare, و VINES أو شبكة أخرى.

- روتينات API للأنبوب المسمّى والشقوق البريديّة في WIN 32. توفّر الأنابيب المسمّاة تداخلاً عالي المستوى لتمرير البيانات بين معالجتين دون إعتبار لجهة كون المعالجة المستقبلة محليّة أو عن بعد والشقوق البريديّة مشابهة، باستثناء أنها عوَضاً عن توفير مسار إتصال واحد مع واحد بين المرسل والمستقبل، توفّر الشقوق البريديّة آليّات إتصال من واحد إلى العديد، ومن العديد إلى واحد. والشقوق البريديّة لبثّ الرسائل إلى أي عدد من المعالجات.
- روتين NetBIOS API: يوفّر هذا الروتين قدرات عكسيّة لتطبيقات MS-DOS و 16-bit Windows و OS/2 التي تمرّر مجاري من البيانات مباشرة عبر الشبكة. كذلك يزوّد إصدار 32-bit جديد.
- روتين Windows Socket: يوفّر روتين API الجديد هذا مقابس 16ـبت و32ـبت وهو تداخل من نوع UNIX قياس لتشبيك الحواسيب. كذلك يوفّر النظام Windows NT طبقات أدنى من الشيفرة تدعم تطبيقات UNIX وتتيح للنظام Windows NT المشاركة بسهولة في شبكة Internet الكبيرة.
- وسيلة إستدعاء إجراء عن بُعد (RPC): تتيح مكتبة وقت التشغيل هذه والمصرّف للمبرمجين الكتابة بسهولة إلى التطبيقات الموزّعة. (راجع القسم 9-4-1 لمزيد من المعلومات).

يجد كل روتين API طريقة إلى الشبكة عبر مسار مختلف يظهر الشكل (9-5) على الصفحة التالية روتينات الدخل / الخرج في Win 32 التي يستخدمها النظام الفرعي Win 32 عن طريق إستدعاء خدمات نظام الدخل / الخرج في NT. بعد ذلك، يصدر برنامج إدارة الدخل / الخرج روتينات IRP إلى مغيّر الوجهة. لكن من الناحية المقابلة، فإن روتينات API في Windows Sockets وفي NetBIOS هي مكتبات DLL التي تستدعي خدمات الدخل / الخرج في NT ويصدر برنامج إدارة الدخل / الخرج حزمات IRP إلى مسيّقات Windows Sockets و NetBIOS على التوالي.

كما يظهر الشكل (9-5)، تمرّر إستدعاءات روتين WNet API (المستخدمة كمكتبات DLL) عبر مكوّن تشبيك يسمى «خدمة محطة العمل». وينسب التعبير خدمة إلى معالجة الملقّم التي توفّر وظيفة معيّنة (يعني وظيفة) وربما تصدير روتين API لدعم تلك الوظيفة. تتضمّن وظائف الخدمة التالي:

- إدارة مغيّر الوجهة الداخلي (خدمة محطة العمل) والملقّم (خدمة الملقّم).
- إرسال رسائل التنبيه إلى المستعملين المسجلين (خدمة المنبّه) ـ مثلاً، عندما يمتلىء القرص الصلب.
- إستلام الرسائل من الأنظمة الأخرى (خدمة الساعي) مثل الإبلاغ عن إتمام وظيفة طباعة.



الشكل (9-5)
الممرّات المختلفة إلى الشبكة

الخدمة هي معالجة مشابهة لنظام فرعي محمي في Windows NT. تشتغل بعض الخدمات في الخلفيّة بينما توفّر الأخرى روتينات API التي يمكن أن تستدعيها شعَب المعالجات الأخرى عن طريق إرسال الرسائل إلى الخدمة. وبعكس الأنظمة الفرعيّة المحميّة، تستعمل الخدمات التي

تزوّد روتينات API عموماً وسيلة تمرير الرسائل RPC عوضاً عن وسيلة LPC للإتصال مع المستضافات. يؤدّي إستعمال RPC إلى جعل الخدمات متوفّرة للمعالجات على الماكنات عن بعد وكذلك للمعالجات المحليّة. (راجع القسم 9-4-1 لمزيد من المعلومات).

إن خدمة محطة العمل هي أساساً «لفاف» نمط المستعمل لمغيّر الوجهة في Windows NT. وهي تنفّذ العمل لدعم روتين WNet API وتوفير وظائف تشكيل مغيّر الوجهة وتحتوي شيفرة نمط المستعمل لإرجاع إحصائيّات مغيّر الوجهة. وعندما يستدعي تطبيق وظيفة WNet API، يمرّر الإستدعاء أولاً إلى خدمة محطة العمل قبل الإنتقال إلى برنامج إدارة الدخل / الخرج NT ومن ثم إلى مغيّر الوجهة.

يكون الكائن المسمّى جهاز التحكّم بالخدمة مسؤولاً عن تحميل خدمات Windows NT وبدئها. وهو أيضاً الوسيلة التي يتمّ بواسطتها تحميل المسيّقات التي لم تحمّل عند وقت الإستنهاض إلى النظام وإلغاء تحميلها منه. يستخدم العديد من مكوّنات التشبيك كمسيّقات وبالتالي فإنها تحمّل في النظام (أو تزال منه) بواسطة جهاز التحكّم بالخدمة.

9-2-2 تشبيك الحواسيب:

رغم شمول العديد من مكوّنات البراجيّات في تشبيك الحواسيب في Windows NT، فإن إثنين من الأكثر أهميّة الّذين لهما التاريخ الأطول عند Microsoft: معيد التوجيه وملقّم الشبكة. وكما في براجيّات MS-NET الأصليّة، يوجّه مغيّر الوجهة طلبات الدخل / الخرج محلياً إلى ملقّم عن بعد، ويستلم الملقّم هذه الطلبات ويعالجها.

وطبعاً، وبإستثناء الأسماء، فالقليل من الأمور العائدة لمغيّر الوجهة أو الملقّم تشبه البراجيّات القديمة. فقد كتبت البراجيّات الأصليّة بلغة assembly وانضفرت حول براجيّات نظام MS-DOS الموجودة. ورغم أن مغيّر الوجهة والملقّم الجديدين مكوّنين داخل Windows NT، فإنها لا تعتمد على تصميم العتاد ونظام التشغيل حيث تعمل وهما مكتوبان باللغة C وبمعنى أن كميّتا نظام ملفّات قابلين للتحميل من النظام في أي وقت. ويمكنها أيضاً التواجد مع معيدات وموجّهات البائعين الآخرين.

يؤدّي إستخدام مغيّر الوجهة والملقّم كمسيّقات نظام ملفّ إلى جعلها جزءاً من البرنامج التنفيذي NT. وبذلك، تتمكّن من الوصول إلى التداخلات الخاصة التي يوفّرها برنامج إدارة الدخل / الخرج للمسيّقات. وقد صمّمت هذه المسيّقات بدورها مع تذكّر متطلّبات مكوّنات الشبكة. يساهم هذا الوصول إلى تداخلات المسيّق إضافة إلى القدرة على إستدعاء وظائف إدارة المخبأ مباشرة إلى حدٍّ كبير في أداء مغيّر الوجهة والملقّم.

يمثّل نموذج المسيّق المرتّب في طبقات لبرنامج إدارة الدخل / الخرج الطبقات الطبيعيّة لبروتوكولات الشبكة. ولأن مغيّر الوجهة والملقّم هما مسيّقان، فإنه يمكن ترتيبهما في طبقات على أعلى قدر ما تريد من مسيّقات بروتوكول النقل. هذه البنية تجعل مكوّنات الشبكة منظومة وتنشىء نقلاً كافياً من طبقة مغيّر الوجهة أو الملقّم نزولاً إلى طبقات النقل والفعليّة للشبكة.

9-2-2-1 مغيّر الوجهة:

يوفّر مغيّر وجهة الشبكة الوسائل الضروريّة لتتمكّن ماكنة واحدة أن تعتمد على Windows NT من الوصول إلى الموارد على الماكنات الأخرى على شبكة. ويستطيع مغيّر الوجهة في Windows NT الوصول إلى الملفّات والأنابيب المسمّاة والطابعات عن بُعد. ولأنه يستخدم بروتوكول SMB، فإنه يعمل مع ملقّمات MS-NET و LAN Manager الموجودة، حيث يتيح الوصول إلى الأنظمة MS-DOS و Windows و OS/2 من Windows NT. تضمن آليات الأمان حماية بيانات Windows NT المشاركة على توصيلة الشبكة من الوصول غير المسموح به.

وكمسيّق لنظام الملفّات، يعمل مغيّر الوجهة كالمسيّقات الأخرى. فعند تحميله في النظام، ينشىء روتين تحفيزه كائن جهاز (يسمّى Device/Rednector/) لتمثيله. كذلك ينشىء روتين التحفيز شيفرات الوظيفة التي تمثّل العمليات التي يتناولها ويسجّل نقاط إدخال المسيّق (روتينات التوزيع) لهذه العمليات. عندما يستلم برنامج إدارة الدخل / الخرج طلب دخل / خرج شبكة، فإنه ينشىء حزمة IRP ويستدعي روتين توزيع في مغيّر الوجهة، حيث يمرّر IRP. بعد أن يعالج مغيّر الوجهة الطلب (بالوصول إلى الشبكة)، تتمّ IRP وترجع النتائج إلى المستدعي.

وبين إرسال طلب وإستلام جواب، يحتوي مغيّر الوجهة على مهمّة أساسيّة واحدة: توفير «نظام ملفّات» يتصرّف كنظام ملفّات محلي رغم أنه يعمل على وسيط غير إعتمادي متأصّل (شبكة). يخفّف الربط الفعلي من حاسوب واحد إلى آخر أكثر مما يخفّف بين حاسوب ومسيّق القرص الصلب أو القرص المرن العائد له. عند إخفاق الربط، يكون مغيّر الوجهة مسؤولاً عن إستعادة التوصيل، حيث أمكن، أو للإخفاق لكي تحاول التطبيقات معاودة العمليّة. يستعمل مغيّر الوجهة الطرق المتنوّعة لتحقيق ذلك. فمثلاً، يعيد بهدوء التوصيل إلى ملقّم عند فقدان التوصيل. ويتذكّر الملفّات التي كانت مفتوحة ويعيد فتحها بظلّ حالات معيّنة. (تحت مغيّر الوجهة، تضمن طبقة النقل الإعتماديّة لنقل البيانات بمستوى البت).

وكسائر مسيّقات نظام الملفّات الأخرى، يجب أن يعمل مغيّر الوجهة ضمن نموذج الدخل / الخرج غير المتزامن، حيث يدعم عمليات الدخل / الخرج غير المتزامنة عند

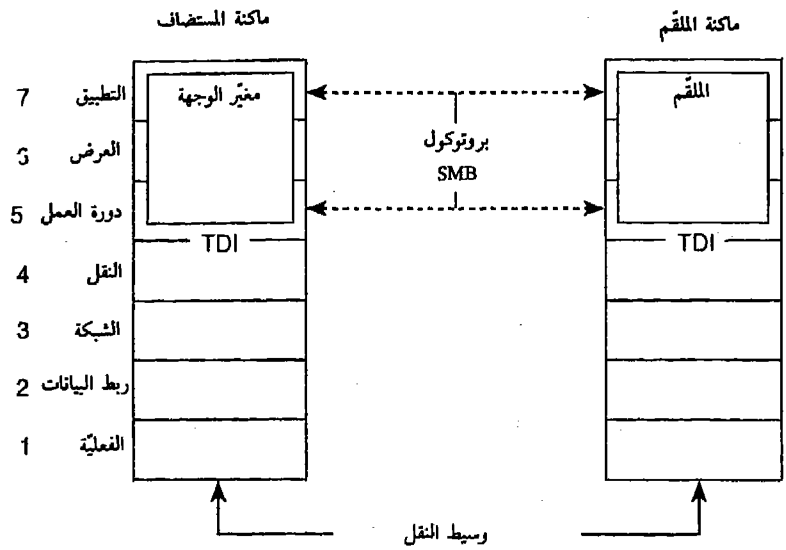
إصدارها. عند إصدار طلب غير متزامن في نمط المستعمل (كما شرح في الفصل الثامن، «نظام الدخل / الخرج»)، يجب أن يرجع مغيّر وجهة الشبكة فوراً في حال إنتهت عمليّة الدخل / الخرج عن بُعد أو لم تنتهي. وفي معظم الحالات، لا ينتهي فوراً طلب دخل / خرج شبكة غير متزامن، لذلك يجب أن ينتظر مغيّر الوجهة إنتهاءه بعد إرجاع التحكُّم إلى المستدعي. وتنشط دائماً شيفرة المسيّق بواسطة شعبة مستدعية ضمن سياق الشعبة. وهي لا تحتوي على فسحة عنوان خاصة بها أو خاصة بأية شعبة. فكيف يطلب المسيّق من روتين الإنتظار؟

هذه المشكلة ليست فريدة بمغيّرات الوجهة، فمعظم مسيّقات أنظمة الملفّات تحتوي على نفس المعضلة. ففي تصميم نظام الدخل / الخرج الأصلي، أنشأت مسيّقات نظام الملفّات التي وجب أن تنفّذ المعالجة ضمن سياقها، معالجة في نمط النواة متعلّقة مع المسيّق واستعملت شعبها لتنفيذ المعالجة غير المتزامنة. لكنه كان حلاً مكلفاً لجهة إستعمال ذاكرة النظام. لذلك، طوّر حلاً جديداً.

يحتوي النظام Windows NT على معالجة نظام خاصة لتحفيز نظام التشغيل عند إستنهاضه. وتحتوي هذه المعالجة على عدّة شعَب عاملة تدور بإنتظار تنفيذ الطلبات نيابة عن المسيّقات والمكوّنات التنفيذيّة الأخرى التي تنفّذ أعمالاً غير متزامنة. فإذا احتاج مسيّق نظام ملفّات إلى شعبة لتنفيذ عمل غير متزامن، فإنه يضع بند العمل في صفيفة لهذه المعالجة الخاصة قبل إرجاع التحكُّم وشيفرة الحالة إلى المستدعي الأصلي. تحفز الشعبة من بروتوكول طبقة التطبيق وتنفّذ العمليات الضروريّة لمعالجة طلب الدخل / الخرج وإتمام حزمة IRP للمستدعي الأصلي.

يرسل مغيّر الوجهة بروتوكولات SMB لتنفيذ عمله. وللتبسيط، يرسم الشكل (9-2) مغيّر الوجهة والملقّم كمكوّنات لطبقة دورة العمل في نموذج تشبيك الحواسيب OSI، فإن بروتوكول SMB هو بروتوكول طبقة تطبيق، كما يوضح ذلك في الشكل (9-6) على الصفحة التالية.

يسمّى التداخل المستعمل من قبل مغيّر الوجهة لإرسال بروتوكولات SMB «تداخل مسيّق النقل» (TDI) يستدعي مغيّر الوجهة روتينات TDI لإرسال بروتوكولات SHB إلى مسيّقات النقل المتعدّدة المحمّلة في Windows NT. ولإستدعاء وظائف TDI، يجب أن يفتح مغيّر الوجهة قناة تسمى «دائرة ظاهريّة» إلى ماكنة المقصد ثم يرسل بروتوكولات SMB على تلك الدائرة الظاهريّة. يحافظ مغيّر الوجهة على دائرة ظاهريّة واحدة لكل ملقّم يوصل إليه النظام Windows NT ويضاعف الطلبات المحدّدة للملقّم عبر نفس الدائرة الظاهريّة. تحدّد طبقة النقل الموجودة تحت مغيّر الوجهة كيفيّة إستخدام الدائرة الظاهريّة فعلياً وإرسال البيانات عبر توصيلة الشبكة.



الشكل (6-9)
إستعمال بروتوكول SMB

صمّم المطوّر Larry Osterman، الذي حوّل مغيّر الوجهة في MS-NET إلى مغيّر الوجهة في MS-DOS LAN Manager 1.0 وإعادة كتابته لمغيّر الوجهة في Lan Manager 2.0، أيضاً مغيّر الوجهة في Windows NT. إضافة لتحسين الإعتماديّة، فقد طلب منه أيضاً توفير توافقيّة كاملة بنسبة 100% مع بروتوكول SMB الموجود. ولتحقيق ذلك، إستعمل نفس البروتوكول لتمرير الرسائل إلى الملقّمات ذات «المستوى المنخفض» (الموجودة في MS-NET و LAN Manager). وتتيح المحافظة على هذا البروتوكول الأساسي للنظام Windows NT التفاعل مع ملقّمات Windows و OS/2 أو MS-DOS التي تشغّل LAN Manager وللنقل بين أنظمة Windows NT، حسّنَ Larry بروتوكول SMB الأساسي لدعم العمليات المشتركة في نظام الدخل / الخرج NT. فمثلاً، يستطيع البروتوكول المحسَّن إستيعاب العمليّات الخاصة للنظام NT مثل فتح ملفّ للوصول إلى الحذف وفتح دليل ووضع لائحات التحكّم بالوصول(ACL) على الملفّات لأغراض الأمان وتنفيذ عمليّات الإستعلام لإسترداد المعلومات المتعلّقة بالملفّات. إضافة لذلك، يمرّر بروتوكول SMB الجديد نضائد النص كأحرف Unicode (أحاديّة الشيفرة) لضمان النقل الصحيح للأحرف من مجموعة الأحرف الدوليّة.

2-2-2-9 الملقّم:

يكتب الملقّم في Windows NT، كمغيّر الوجهة، لتوافقيّة كاملة بنسبة 100% مع بروتوكولات MS-NET و LAN Manager الموجودة. تتيح هذه التوافقيّة الكاملة للملقّم معالجة

الطلبات الصادرة من أنظمة Windows NT وأيضاً تلك الصادرة عن الأنظمة الأخرى التي تشغّل براجيّات LAN Manager. ويستخدم الملقّم، كمغيّر الوجهة، كمسيّق لنظام الملفّات.

وقد يتساءل أحدهم لماذا لا يستخدم شيء يسمّى «ملقّم» كمعالجة ملقّم قد يتوقّع أن يعمل ملقّم شبكة كنظام فرعي محمي ـــ وهي معالجة ذات شعَب تنتظر وصول الطلبات من الشبكة، وتنفيذها، ثم إرجاع النتائج إلى الشبكة. وهذا الخيار هو الأرجح، وقد إعتبره Chuck Lenzmeier بعناية عندما بدأ بتصميم الملقّم في Windows NT، فقد قرّر Chuck، المطوِّر الرئيسي للملقّم وخبير تشبيك الحواسيب المعتمد على VAX/VMS وRPC، أن يستخدم الملقّم كمسيّق لنظام الملفّات. ورغم عدم كون الملقّم مسيّق بالمعنى الحقيقي ورغم أنه لا يدير نظام الملفّات، فإن إستعمال نموذج المسيّق يوفّر ميزات عديدة على إستخدام الملقّم كمعالجة.

فالميزة الرئيسيّة هي عند إستعماله كمسيّق، يتواجد الملقّم ضمن البرنامج التنفيذي NT ويمكنه إستدعاء برنامج إدارة المخبأ NT مباشرة لإستمثال عمليات نقل بياناته. فمثلاً، إذا إستلم الملقّم طلب قرءاة كميّة كبيرة من البيانات، فإنه يستدعي برنامج إدارة المخبأ ليحدّد موقع البيانات في المخبأ (أو تحميل البيانات في المخبأ إذا لم تكن موجودة هناك) ولقفل البيانات في الذاكرة. بعد ذلك، ينقل الملقّم البيانات مباشرة من المخبأ إلى الشبكة، حيث يتجنّب عمليات الوصول إلى القرص ونسخ البيانات غير الضروريّة. وبشكل مشابه، وفي حال طلب منه كتابة البيانات، يستدعي الملقّم برنامج إدارة المخبأ ليعيد تلقيم النسخة للبيانات القادمة وتعيين موقع مخبأ لها. بعد ذلك، يكتب الملقّم البيانات مباشرة في المخبأ. وعن طريق الكتابة إلى المخبأ عوَضاً عنه إلى القرص، يستطيع الملقّم إرجاع التحكُّم إلى المستضاف بسرعة. بعد ذلك يكتب برنامج إدارة المخبأ البيانات إلى القرص في الخلفيّة (بإستعمال براجيّات الترتيب في صفحات لبرنامج إدارة VM).

عند إستدعاء برنامج إدارة المخبأ، يفترض الملقّم فعلياً بعض مسؤوليّات برنامج إدارة الدخل / الخرج لتحقيق معالجة أكثر تنظيماً. والطريقة الأخرى التي يفترض فيها الملقّم هذا الدور، هي عند تنسيق حزمات IRP الخاصة به ويمرّرها مباشرة إلى مسيّقات NTFS وFAT وHPFS. ويمكنه إختيار نسخ البيانات إلى المخبأ ومنه مباشرة عوَضاً من إنشاء حزمات IRP. وإذا كان نظاماً فرعياً في نمط المستعمل (أو حتى في نمط النواة)، فإنه يستدعي عوَضاً عن ذلك خدمات دخل / خرج NT لمعالجة الطلبات الداخلة، الأمر الذي يتطلّب كلفة إضافيّة.

وكمسيّق نظام ملفّات، يحتوي الملقّم أيضاً على مرونة أكثر ما يحتويه عندما يكون معالجة. فمثلاً، فإنه يستطيع تسجيل روتين إتمام دخل / خرج، الذي يتيح إستدعاءه فوراً بعد أن تنتهي المسيّقات لطبقة منخفضة من معالجتها، ولكي يتمكّن من تنفيذ أية معالجة لاحقة مطلوبة. ورغم

إستخدام الملقّم في Windows NT كمسيّق نظام الملفّات، يمكن إستخدام الملقّمات الأخرى إما كمسيّقات أو كمعالجات ملقّم.

يستعمل الملقّم، كمغيّر الوجهة، معالجة النظام الأوليّة لمناولة عمليّات الدخل / الخرج غير المتزامنة والعمليّات بكلفة إضافيّة مرتفعة المستعملة أحياناً والتي لا تتطلّب سرعة مثلى، مثل فتح ملفّ. أما العمليّات التي تتطلّب سرعة قصوى، مثل القراءة والكتابة، فهي تنفّذ مباشرة في المسيّق عند الإمكان لتجنب الكلفة الإضافيّة لتحويل سياقي. ورغم أنها تحتاج لتحويل سياقي، فإن إستعمال معالجة النظام الأوليّة لتنفيذ عمليّات الشبكة يتيح للملقّم إنتظار مقابض كائن عند الضرورة أو إستعمال أخطاء الصفحة، ولا يمكن إستعمال أي منها عند التنفيذ في سياق شعبة أخرى عند مستوى IRQL مرتفع. إن القدرة على إستعمال أخطاء الصفحة يعني أنه يتوجّب على نسبة أقلّ من شيفرة الملقّم البقاء في الذاكرة.

عند إستنهاض النظام Windows NT، يحمّل برنامج إدارة الدخل / الخرج هذه المسيّقات المطلوبة سابقاً في تتابع الإستنهاض (مثل مسيّقات نظام الملفّات والقرص، ومسيّق الفيديو ومسيّقات الماوس ولوحة المفاتيح). وبعد أن يصل تتابع الإستنهاض إلى نقطة التحويل من نمط النواة إلى نمط المستعمل، تحفّز وحدة التحكُّم بالخدمة لتحميل بقيّة المسيّقات، بما فيها مغيّر الوجهة والملقّم ومسيّقات النقل للشبكة. تحمّل وحدة التحكُّم بالخدمة هذه المسيّقات عن طريق إستدعاء خدمات نظام الدخل / الخرج التي تنسخها إلى الذاكرة وتنفّذ روتينات تحفيزها. ينشىء كل روتين تحفيز للمسيّق كائن جهاز ويدرجه في فسحة إسم برنامج إدارة الكائنات. ويمكن تحفيز وحدة التحكُّم بالخدمة أيضاً في أي وقت بعد إشتغال النظام لتحميل أو إلغاء تحميل ملقّمات الشبكة أو لإيقاف أو بدء خدمات النظام.

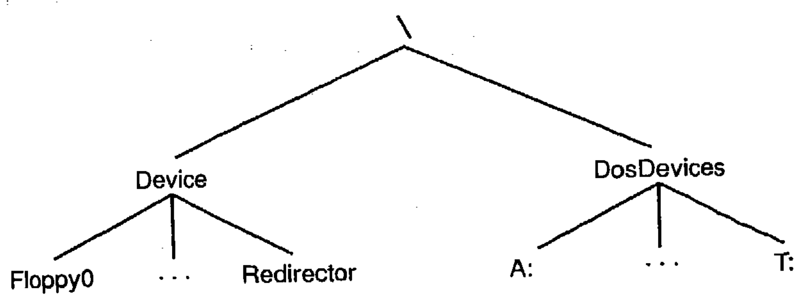
3-2-2-9 دقّة الإسم:

إحدى الأهداف الرئيسيّة التي حقّقها النظام Windows NT هي في تمديد بلوغ نظام الدخل / الخرج المحلي ليشمل الموارد عن بُعد. ولأن كل مثل هذه الموارد هي كائنات، فإنه يتوجّب على براجيّات تشبيك الحواسيب العمل ضمن بنية الكائن المحليّة للوصول إلى هذه الموارد. وعندما يفتح تطبيق ملفّ، فإنه يفتح فعلياً مقبضاً إلى كائن ملفّ NT. يحتوي كائن الملفّ البيانات الخاصة بتلك «الحالة الآنية المفتوحة» للملفّ ـ مثلاً، بيانات نمط المشاركة ومؤشّر الملفّ. وتكون المعالجة هي نفسها عندما يقع الملفّ الواجب فتحه على حاسوب من بعد. ويتدخّل برنامج إدارة الكائنات عن طريق إنشاء كائن ملفّ وفتح مقبض إليه.

وكما هي الحال مع الملفّات المحليّة، يجب أن يكون إسم الملفّ عن بعد، الذي يفتحه

التطبيق، دقيقاً. أي، يجب أن يحدّد نظام التشغيل الجهاز حيث يوجد الملفّ، ونظام الملفّات المستعمل على الجهاز وموقع الملفّ ضمن نظام الملفّات. وللملفّات عن بعد، يجب أن يتأكّد نظام التشغيل من الماكنة حيث يخزّن الملفّ ومن كيفيّة إرسال طلب إلى تلك الماكنة.

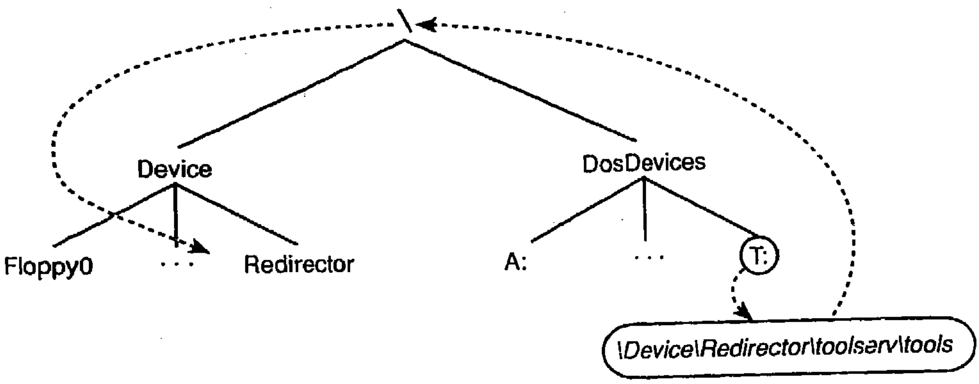
إفترض أن المستعمل عيّن حرف سوّاقة إلى ملقّم عن بعد، عن طريق إصدار الأمر NET USE T: \ TOOLSERV \ TOOLS. تنشىء خدمة محطة العمل كائن ربط رمزي يسمى :T في فسحة إسم برنامج إدارة الكائنات NT، كما يبيّن في الشكل (9-7).



الشكل (9-7)
فسحة إسم برنامج إدارة الكائنات

ولاحقاً، يفتح تطبيق Win 32 الملفّ عن بعد T: \ editor.exe. يترجم النظام الفرعي Win 32 الإسم إلى كائن NT ,T: \ editor.exe \ DosDevice \ ويستدعي البرنامج التنفيذي NT لفتح الملفّ. وخلال المعالجة، يكتشف برنامج إدارة الكائنات أن :T \ DosDevice \ هو كائن ربط رمزي ويستبدل الإسم المحدّد للكائن :T \ DosDevice \. وكما يبيّن في الشكل (9-8)، فإنّ Redirector \ Device \ هو إسم كائن الجهاز الذي يمثّل مغيّر الوجهة في Windows NT ويشير :T إلى مشاركة LAN Manager عن بعد التي سيحدّد موقعها مغيّر الوجهة.

تستعمل كائنات الجهاز في NT كنقطة الإطلاق إلى فسحة إسم كائن غير محكوم من قبل برنامج إدارة الكائنات NT. وعند تحليل إسم كائن نحوياً، وإذا وجد برنامج إدارة الكائنات كائن جهاز في المسار، فإنه يستدعي طريقة التحليل النحوي المتعلّقة بكائن الجهاز. في هذه الحالة، تكون الحالة عبارة عن روتين برنامج إدارة دخل / خرج يستدعي مغيّر الوجهة. يبني مغيّر الوجهة بروتوكولات SMB ويمرّرها عبر مسيّق النقل إلى ملقّم SMB عن بعد الذي يفتح



الشكل (9-8)
التدقيق في إسم ملفّ شبكة

الملفّ editor.exe \ على TOOLS \ TOOLSERV \. ينشىء برنامج إدارة الكائنات NT كائن ملفّ محلي لتمثيل الملفّ المفتوح وإرجاع مقبض كائن إلى المستدعي. بعد ذلك، تمرّر عمليّة على مقبض الكائن مباشرة إلى مغيّر الوجهة في Windows NT.

توجد فسحة إسم كائن مشابهة على نظام المقصد في Windows NT. تحتوي فسحة إسم الكائن عن بعد الإسم Server \ Device \ الذي يشير إلى مسيّق نظام الملفّات الذي يستخدم وظائف الملقّم الداخليّة في Windows NT. لكن، لا يستعمل كائن الجهاز هذا عندما يستلم الملقّم طلباً. وهو يستعمل فقط عندما يشير مدير نظام إلى الملقّم بإسمه خلال إدارته.

3-9 التصميم المنفتح:

لأنّ ملقّم الشبكة مركّب داخل النظام Windows NT، قد يبدو وكأن الملقّم موصّل بأسلاك، ــ لكن ذلك غير صحيح. حدّد هدف «التركيب والتشغيل الفوري» من قبل Dave Thompson أن النظام Windows NT يستطيع الربط إلى مجموعة متنوّعة من الشبكات. لذلك، وإضافة إلى إمكانيّة تحميل مسيّقات مغيّر الوجهة والملقّم والنقل، وإلغاء تحميلها ديناميأ، فإنه يمكن لمجموعة متنوّعة من هذه الكائنات التواجد أيضاً. فالنظام Windows NT يدعم شبكات أخرى إضافة إلى LAN Manager بعدّة طرق:

- يوفّر الوصول إلى أنظمة ملفّات غير عائدة إلى Microsoft لتصفّح الشبكة وتؤجّل الموارد وللملفّ عن بعد ودخل / خرج الجهاز عبر روتين Win 32 API عام (روتين WNet API).

- يتيح تحميل مسيّقات بروتوكول نقل الشبكة المتعدّدة في نفس الوقت ويتيح لمغيّرات الوجهة إستدعاء تداخل عام واحد للوصول إليها.
- يزوّد تداخل ومحيط (NDIS 3.0) لمسيّقات بطاقة الشبكة للوصول إلى مسيّقات النقل في Windows NT ولكسب النقليّة إلى أنظمة MS-DOS المستقبليّة.

تعالج الأقسام التالية كلاً من هذه القدرات والبراجيّات المستعملة لإستخدامها.

1-3-9 الوصول في نمط المستعمل إلى أنظمة الملفّات عن بُعد:

كما ذُكر في القسم 9-2-1، توفّر روتينات API للدخل / الخرج وWin 32 WNet طريقتين للتطبيقات في نمط المستعمل من الوصول إلى الملفّات (والموارد الأخرى) على الأنظمة عن بُعد. يستعمل روتينا API قدرات مغيّر الوجهة لإيجاد سبيلها إلى الشبكة. ورغم أن الشروح السابقة ركّزت على براجيّات الشبكة الداخليّة، فإنه يمكن تحميل مغيّرات وجهة إضافيّة في النظام للوصول إلى الأنواع المختلفة من الشبكات. يوسّع هذا القسم المثال الأصلي عن طريق شرح البراجيّات التي تحدّد مغيّر الوجهة الواجب تحفيزه عند إصدار طلبات الدخل / الخرج عن بُعد. أما المكوّنات المسؤولة فهي:

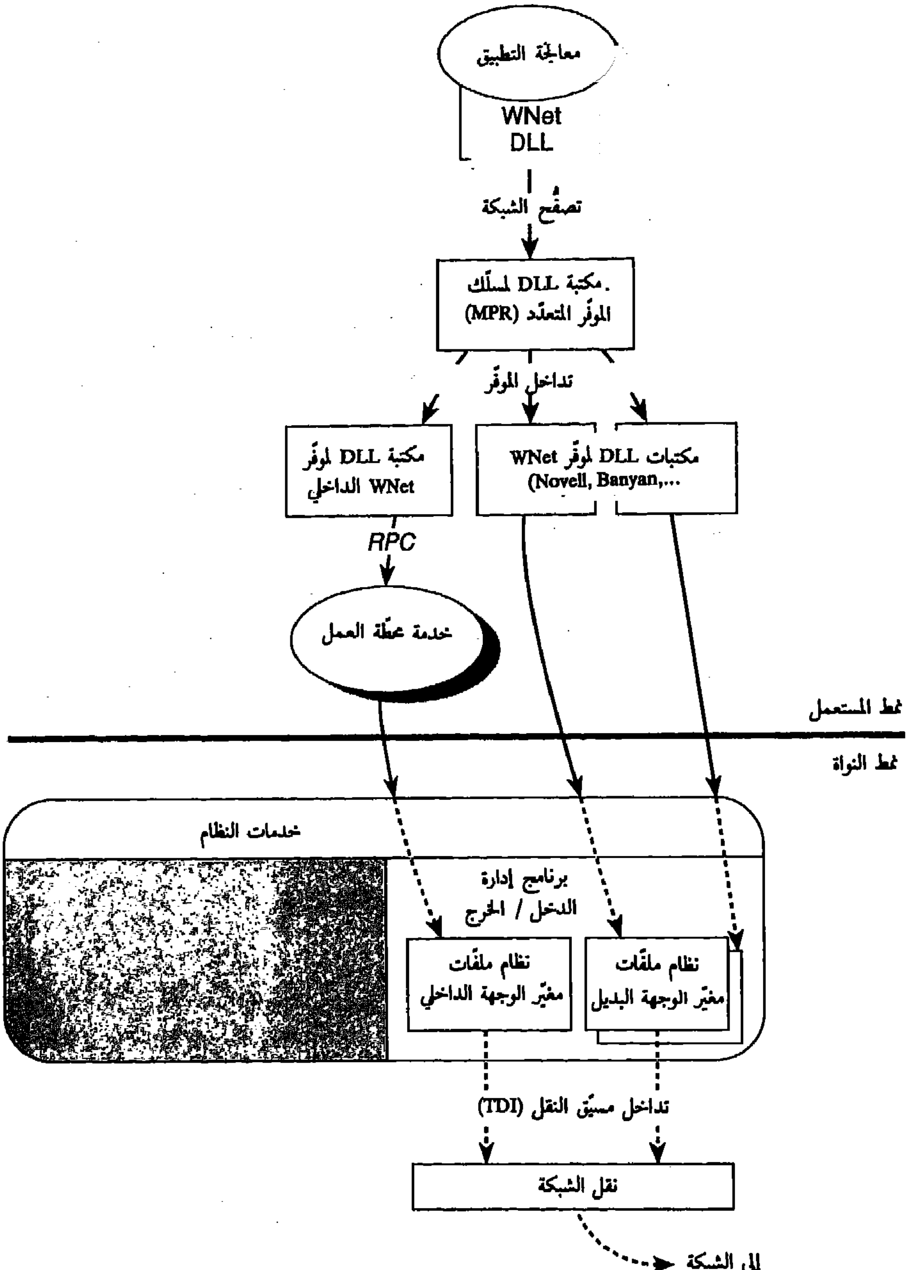
- مسلّك الموفّر المتعدّد (MPR). مكتبة DLL تحدّد الشبكة الواجب الوصول إليها عندما يستعمل تطبيق روتين Win 32 API لتصفّح أنظمة الملفّات عن بُعد.
- موفّر UNC المتعدّد (MUP). مسيّق يحدّد الشبكة الواجب الوصول إليها عندما يستعمل تطبيق روتين Win 32 API لتصفّح أنظمة الملفّات عن بُعد.

1-1-3-9 مسلّك الموفّر المتعدّد لروتين WNet API:

تتيح وظائف Win 32 WNet للتطبيقات (بما فيها برنامج إدارة الملفّات في Windows) التوصيل إلى موارد الشبكة، مثل ملقّمات الملفّ والطابعات وتصفّح محتويات أي نوع من نظام الملفّات عن بُعد. ولأنه يمكن إستدعاء الروتين API للعمل عبر شبكات مختلفة بإستعمال بروتوكولات نقل مختلفة، يجب أن تكون البراجيّات موجودة لإرسال الطلب بشكل صحيح عبر الشبكة ولإستيعاب النتائج التي يرجعها الملقّم عن بُعد. يُظهر الشكل (9-9) على الصفحة التالية البراجيّات المسؤولة عن هذه المهمّة.

الموفّر هو براجيّات تجعل Windows NT مستضافاً لملقّم شبكة عن بُعد. تشمل بعض العمليّات المنفّذة من قبل موفّر WNet، إنشاء توصيلات الشبكة وقطعها والطباعة عن بُعد ونقل

البيانات. ويشمل موفّر WNet الداخلي مكتبة DLL وخدمة محطة العمل ومغيّر الوجهة الداخلي. وقد يزوّد بعض بائعي الشبكات فقط مكتبة DLL ومغيّر وجهة.

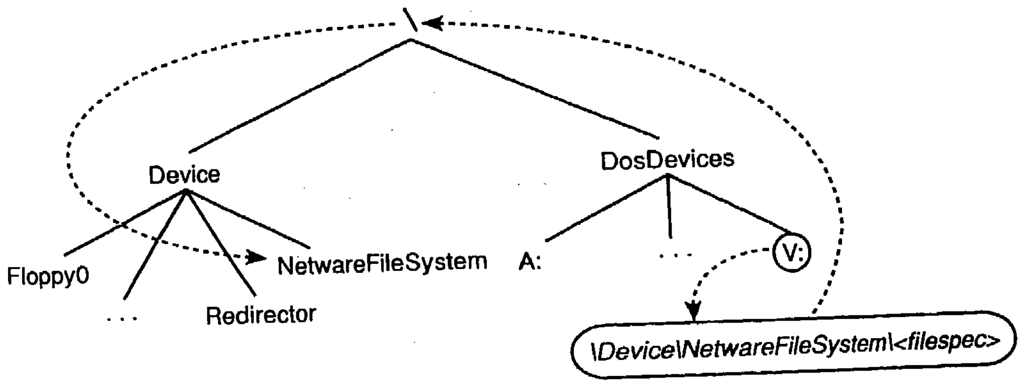


الشكل (9-9)
براجيّات الموفّر المتعدّد

عندما يستدعي تطبيق روتين WNet، يمرّر الإستدعاء مباشرة إلى مكتبة مسلّك الموفّر المتعدّد (MPR) وهي مكوّن تشبيك مصمّم من قبل Chuck Chan. يستلم المسلّك MPR الإستدعاء ويحدّد موفّر WNet الذي يتعرّف إلى الموارد التي يحاول الوصول إليها. وتزوّد كل مكتبة DLL لموفّر تحت المسلّك MPR مجموعة من الوظائف القياسيّة جماعياً والتي تسمّى «تداخل الموفّر». يتيح هذا التداخل للمسلّك MPR تحديد الشبكة التي يحاول التطبيق الوصول إليها وتوجيه الطلب إلى براجيّات موفّر WNet المناسب.

عند إستدعائه من قبل روتين WNetAddConnection () API للتوصيل إلى مورد شبكة عن بُعد، يدقّق المسلّك MPR بسجلّ التشكيك لتحديد موفّرات الشبكة المحمّلة. وهو يسحبها واحدة بعد الأخرى بالترتيب المسرّدة فيه في السجل إلى أن يتعرّف مغيّر الوجهة إلى المورد أو إلى أن يتمّ سحب كل الموفّرات المتوفّرة. (يمكن أيضاً تغيير الترتيب عن طريق تحرير بيانات السجل).

يستطيع الروتين WNetAddConnection () API أيضاً تعيين حرف سوّاقة أو إسم جهاز إلى مورد عن بعد. وعند الطلب منه للقيام بذلك، يسلّك الروتين WNetAddConnection () الإستدعاء إلى موفّر الشبكة المناسب. وينشىء الموفّر، بدوره، كائن ربط رمزي NT يخطّط حرف السوّاقة المعرّف إلى مغيّر الوجهة (أي، مسيّق نظام الملفّات عن بُعد) لتلك الشبكة. يوضح الشكل (9-10) كيفيّة ملاءمة أسماء موارد الشبكة في فسحة إسم برنامج إدارة الكائنات NT.



الشكل (9-10)
التدقيق في إسم مورد شبكة

تنشىء مغيّرات الوجهة الأخرى، كمغيّر الوجهة الداخلي، كائن جهاز في فسحة إسم برنامج إدارة الكائنات عند تحميلها في النظام وبعد تحفيزها. بعد ذلك، وعندما يستدعي روتين WNet أو روتين API آخر برنامج إدارة الكائنات لفتح مورد على شبكة مختلفة، يستعمل برنامج إدارة الكائنات كائن الجهاز كنقطة الإنطلاق إلى نظام الملفّات عن بُعد. ثم يستدعي طريقة التحليل النحوي لبرنامج إدارة الدخل / الخرج المتعلّقة بكائن الجهاز لتحديد موقع مسيّق نظام ملفّات مغيّر الوجهة الذي يستطيع مناولة الطلب. (راجع الفصل الثامن، «نظام الدخل / الخرج» لمزيد من المعلومات).

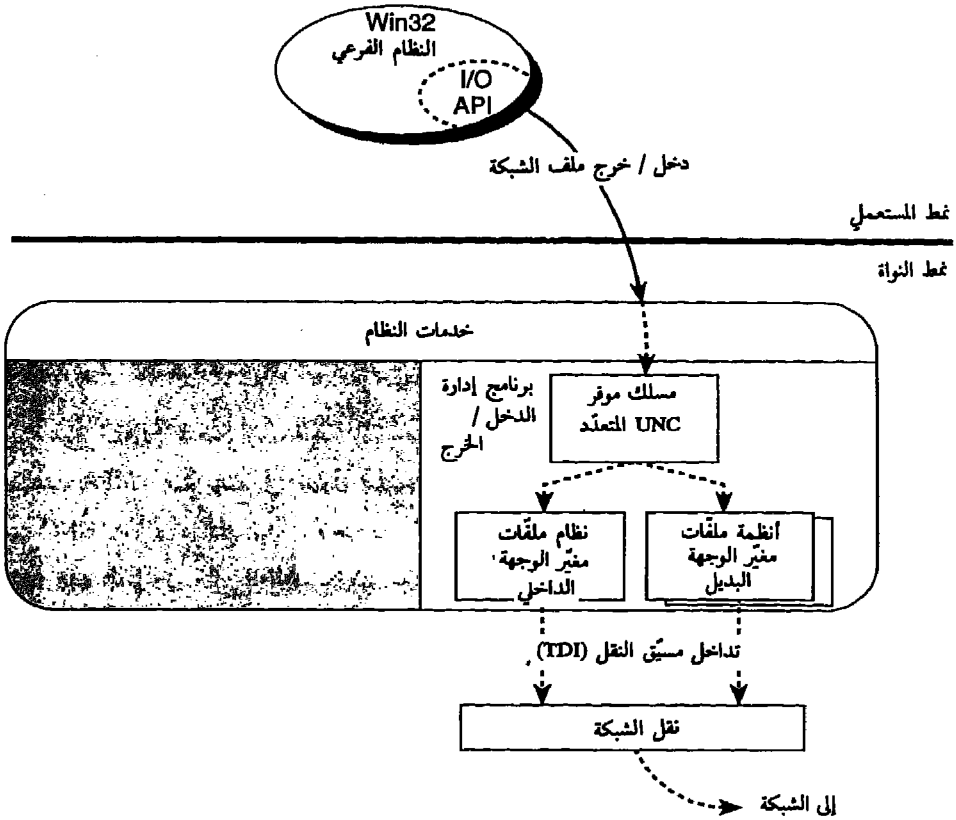
2-1-3-9 موفّر UNC المتعدّد لدخل / خرج الملفّ في Win 32:

موفّر UNC المتعدّد (MUP)، المصمّم من قبل Manny Weiser، هو مكوّن تشبيك مشابه للمسلّك MPR. فهو يضع في حقول طلبات الدخل / الخرج المحددة لملفّ أو جهاز يحتوي على إسم UNC (الأسماء التي تبدأ بالأحرف\\، والتي تشير إلى وجود المورد على الشبكة) يستلم MUP هذه الطلبات مثل MPR، ويحدّد مغيّر الوجهة المحلي الذي يتعرّف إلى المورد عن بُعد.

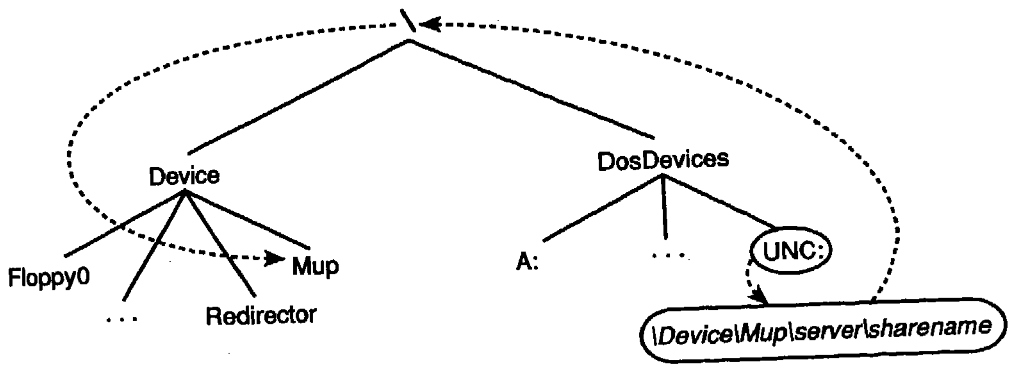
وبعكس MPR، فإن MUP هو مسيّق NT (محمّل في وقت إستنهاض النظام) يصدر طلبات الدخل / الخرج إلى المسيّقات بطبقة منخفضة، وفي هذه الحالة إلى مغيّرات الوجهة، كما يبيّن في الشكل (9-11).

ينشّط المسيّق MUP عندما يحاول تطبيق لأول مرة فتح ملفّ أو جهاز عن بُعد، حيث يحدّد إسم UNC (عوَضاً عن حرف سوّاقة مغيّر وجهة، كما وصف سابقاً في هذا الفصل). وعندما يستلم النظام الفرعي Win 32 مثل هذا الطلب، يلحق النظام الفرعي إسم UNC إلى النضيد DosDevices\UNC\ ثم يستدعي برنامج إدارة الدخل / الخرج NT لفتح الملفّ. وإسم ملفّ الكائن هذا هو إسم ربط رمزي يعود إلى Device\Mup\ / الخرج.

يستلم مسيّق MUP الطلب ويرسل حزمة IRP غير متزامنة إلى كل مغيّر وجهة مسجّل. بعد ذلك، ينتظر إحداها للتعرُّف إلى إسم المورد والإجابة. وعندما يتعرّف مغيّر وجهة إلى الإسم، فإنه يشير إلى مدى كون الإسم فريداً له. فمثلاً، إذا كان الإسم HELENC\\ PUBLIC\ inside\scoop.doc\، يتعرّف مغيّر الوجهة في Windows NT إليه ويدّعي أن النضيد HELENC\PUBLIC\\ له. يخبىء المسيّق MUP هذه المعلومات ثم يرسل الطلبات بدءاً من ذلك النضيد مباشرة إلى مغيّر الوجهة في Windows NT، حيث يتجاوز عمليّة «السحب». ويتّصف مخبأ مسيّق MUP بميزة نفاذ الوقت، بحيث ينفّذ وقت النضيد المتعلّق بمغيّر وجهة معيّن بعد فترة زمنيّة من عدم الإستعمال.



الشكل (9-11)
موفّر UNC المتعدّد (MUP)



الشكل (9-12)
التدقيق في إسم UNC

إذا إدّعى أكثر من مغيّر وجهة واحد حصوله على مورد معيّن، يستعمل المسيّق MUP لائحة مغيّرات الوجهة المحمّلة لمسجّل التشكيل لتحديد مغيّر الوجهة الأسبق. يمكن إعادة ترتيب لائحة مغيّرات الوجهة عن طريق تحرير قاعدة بيانات المسجّل.

2-3-9 بروتوكولات النقل:

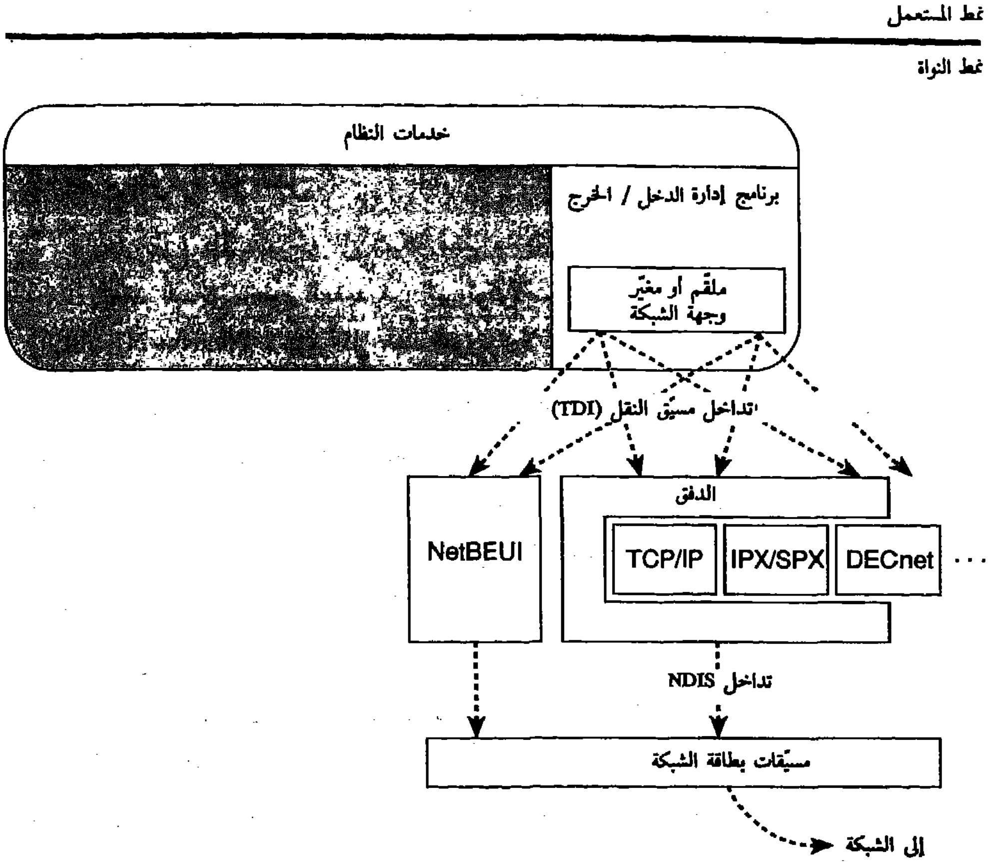
بعد بلوغ طلب موجّه إلى شبكة مغيّر وجهة، يجب أن يسلّم الطلب إلى الشبكة. ففي خلال العقد السابق، طوّر العديد من البروتوكولات المختلفة لإرسال المعلومات عبر الشبكات. والنظام Windows NT لا يوفّر كل هذه البروتوكولات، لكنّه يتيح توفيرها.

تستخدم بروتوكولات النقل في Windows NT كمسيّقات والتي يمكن تحميلها كسائر مغيّرات الوجهة والملقّمات، في النظام ومنه. وفي نموذج تشبيك حواسيب إصطلاحي، يجب على مغيّر وجهة يستعمل بروتوكول نقل معيّن أن يعرف نوع الدخل الذي يتوقّعه مسيّق البروتوكول وأن يرسل الطلبات إليه في هذا النسق، يجب إعادة كتابة الطبقات السفليّة لمغيّر الوجهة لدعم آليات البيانات المختلفة لكل آلية نقل مستعملة.

يتجنّب النظام Windows NT هذه المشكلة عن طريق توفير تداخل برمجة واحد ـــ يسمى تداخل مسيّق النقل (TDI) ـ لمغيّرات الوجهة ومسيّقات الشبكة بمستوى إرتفاع الأخرى. يتيح TDI لمغيّرات الوجهة والملفّات البقاء مستقلّة عن آليات النقل. وهكذا يستطيع إصدار واحد لمغيّر وجهة أو ملقّم إستعمال أية آلية نقل متوفّرة كما يبيّن في الشكل (9-13).

إن التداخل TDI هو تداخل مستقلّ عن آلية النقل غير متزامن يستخدم آلية عنونة وراثيّة ومجموعة متنوّعة من الخدمات والمكتبات. يوفّر كل مسيّق نقل التداخل في طبقته الأعلى لكي تستطيع مغيّرات الوجهة (والملقّمات على ماكنات Windows NT عن بُعد) إستدعاءه دون إعتبار لآلية النقل قيد الإستعمال تحت التداخل. ولإرسال طلب، يستدعي برنامج إدارة الدخل / الخرج مغيّر وجهة حيث يمرّر إليه حزمة IRP للمعالجة. يتناول مغيّر الوجهة الداخلي هذا الطلب عن طريق تمرير بروتوكولات SMB عبر وصلة دائرة ظاهريّة إلى ملقّم عن بُعد. وتستطيع مغيّرات الوجهة الأخرى إستعمال وسائل أخرى للإتصال مع الملقّمات عن بُعد.

يوفّر التداخل TDI مجموعة من الوظائف التي تستطيع مغيّرات الوجهة إستعمالها لإرسال أي نوع من البيانات عبر آلية نقل. ويدعم التداخل TDI الإرسالات المعتمدة على الوصلة (الدائرة الظاهريّة) والإرسالات دون وصلة (مخطّط البيانـات). ورغم أن LAN Manager يستعمل الإرسالات المعتمدة على وصلة، غير أن براجيّات Novell's IPX هي مثال عن شبكة تستعمل الإتصالات دون وصلة. تزوّد Microsoft آليّات النقل التالية:



الشكل (13-9)
تداخل مسيّق النقل (TDI)

■ آليّة النقل NetBEUI (التداخل الموسّع مع المستعمل NetBIOS Extended User Interface).
NetBEUI هو بروتوكول نقل مناطقي محلي طوّر من قِبَل IBM ليعمل بظلّ تداخل شبكة NetBIOS من Microsoft.

■ آليّة النقل TCP/IP (بروتوكول التحكّم بالإرسال / بروتوكول مجموعة الشبكات Internet).
TCP/IP هو بروتوكول طوّر لوزارة الدفاع الوطنيّة الأميركيّة لوصل الأنظمة المتغايرة العناصر على شبكة مناطقيّة واسعة. تستعمل TCP/IP عموماً في شبكات UNIX وتتيح للنظام Windows NT المشاركة في لوحات الإعلان والأخبار والخدمات البريديّة الإلكترونيّة التي تعتمد على UNIX. تعمل آليّة النقل TCP/IP في محيط متوافق مع STREAMS.

تتضمّن آليّات النقل الأخرى الموجودة أو قيد التطوير من قبَل Microsoft أو البائعين الآخرين:

- IPX/SPX (تبادل حزمات البيانات بين الشبكات / تبادل حزمات البيانات التتابعي Sequenced Packet Exchange/Internet Packet Exchange).
  IPX/SPX هو مجموعة من بروتوكولات النقل المستعملة من قبل براجيّات Netware من قبل شركة Novell Corporation.
- آليّة النقل DECnet. DECnet هو بروتوكول متملّك مستعمل من قبل شركة Digital Equipment Corporation ويزوّد لربط أنظمة Windows NT إلى شبكات DECnet.
- آليّة النقل Apple Talk. هو بروتوكول مطوّر من قبل شركة Apple Computers Inc. يتيح لأنظمة Apple Macintosh الإتصال مع Windows NT.
- آليّة النقل XNS (أنظمة الشبكات من XeroxNetwork Systems). XNS هو بروتوكول نقل مطوّر من قبل شركة Xerox Corporation إستعمل سابقاً في شبكات Ethernet.

ويجدر هنا شرح إضافي حول المحيط STREAMS. فهو محيط تطوير مسبق في النظام في V من UNIX والذي يتيح لمسيقات النقل تحقيق درجة عالية من النقلّية من نظام تشغيل واحد إلى آخر. يتيح المحيط STREAMS (الذي يخطط إلى التداخل TDI عند حدوده العليا وإلى NDIS 3.0 عند حدوده السفلى) للعديد من مسيقات النقل الموجودة والتي تعتمد على STREMS من التوصل إلى Windows NT مع تعديل بسيط أو حتى دون تعديل. ويمكن استخدام مسيقات النقل مثل IPX/SPX و DECnet والأخرى إما كمسيقات STREAMS أو كمسيقات أحادية الطبقة (مثل NetBEUI).

3-3-9 محيط NDIS لمسيقات الشبكة

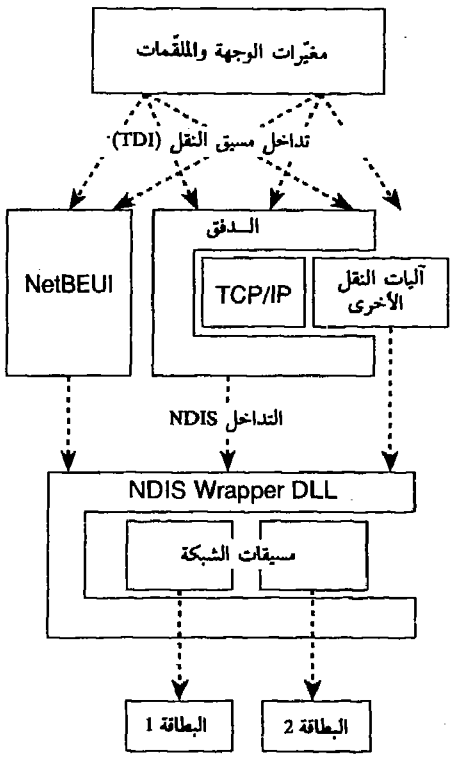
لا تضع مسيقات البروتوكول التي سبق وشرحت في القسم السابق البتات الفعلية على الشبكة. وتتوفر وصلة الكابل من قبل بطاقة شبكة أو شريحة تكون إما داخلية أو تزلّق في شق الإضافة في مؤخرة الحاسوب. تستطيع كل بطاقة شبكة (تسمى أحيانا مهايىء شبكة) في الاتصال بواسطة نوع معيّن من الكابلات، باستعمال طبولوجيا شبكة معيّنة.

تتوفر بطاقات الشبكة أحياناً مع مسيقات الشبكة والتي استخدمت غالباً في السابق IP. ولأن النظام Windows NT يتيح تحميل العديد من مسيقات البروتوكولات المختلفة، قد يرغب كل بائع لبطاقات الشبكة هذه يستعمل هذه الطريقة من إعادة كتابة مسيقاته لتدعم البوتوكولات المتنوعة ــ وهذه ليست استراتيجية مثالية. ولمساعدة البائع على تجنّب هذا العمل

غير الضروري، يوفّر النظام Windows NT تداخلاً ومحيطاً يسمى مواصفات التداخل لمسيق الشبكة (NDIS) الذي يحمي مسيقات الشبكة من تفاصيل بروتوكولات النقل المتنوعة والعكس بالعكس. يوضح ذلك الشكل (9-14).

وعوضاً عن كتابة مسيق خاص للنقل للنظام Windows NT، يوفّر بائعو الشبكات التداخل NDIS على أنه الطبقة الأعلى لمسيق شبكة واحد. وبذلك، فإنهم يتيحون لأي مسيق بروتوكول من توجيه طلبات شبكته إلى بطاقة الشبكة عن طريق استدعاء هذا التداخل. وهكذا، يستطيع المستعمل الاتصال بواسطة شبكة TCP/IP وشبكة NetBEUI (أوDECnet و NetWare و VINES وما شابه) باستعمال بطاقة شبكة واحدة ومسيق شبكة واحد.

التداخل NDIS متوفّر في LAN Manager لكنه حدّث في Windows NT إلى الاصدار 3.0. فالاصدار 3.0 نقّال (مكتوب باللغة C) ومحدّث ليستعمل عناوين من 32 بت عوضاً عن عناوين



الشكل (9-14)
التداخل NDIS 3.0

من 16 بت ويمكّن بواسطة معالـج متعدد. وكالإصدارات السابقة، فإنه يستطيـع مناولة وصلات شبكة مستقلة متعددة وبروتوكولات نقل متعددة محمّلة في نفس الوقت.

ويكون كل مسيق شبكة NDIS مسؤولاً عن إرسال الحزمات واستلامها على وصلة الشبكة وعن إدارة البطاقة الفعلية نيابة عن نظام التشغيل. وعند حدوده السفلى، يتصل مسيق NDIS مباشرة مـع البطاقة أو البطاقات التي يجذفها باستعمال روتينات NDIS للوصول إليها. يبدأ مسيق NDIS الدخل / الخرج على البطاقات ويستلم المقاطعات منها. وهو يستدعي مسيقات البروتوكول للإشارة إلى استلامه البيانات ولإبلاغها إتمامه نقل البيانات الخارجة.

يتيـح NDIS لمسيقات الشبكة أن تكون نقّالة دون معرفتها الضمنية للمعالـج أو نظام التشغيل حيث تشتغل. وتستطيـع مسيقات الشبكة استدعاء روتينات NDIS لحماية نفسها من المعلومات الخاصة بالمنصة بحيث تستطيـع التحرك بسهولة من نظام Windows NT إلى آخر أو من النظام Windows NT إلى أنظمة Windows / MS-DOS مستقبلية. وفي Windows NT، تستدعي براجيات NDIS روتينات النواة NT للحصول على الأقفال الدواميّة وإفلاتها (للتشغيل الآمن للمعالـج المتعدد) وتستدعي روتينات برنامـج إدارة الدخل / الخرج لوصل كائنات المقاطعة إلى مستوى IRQL المناسب في جدول توزيـع النواة. وهذان مثالان فقط عن المهام التي يمكن أن يقوم بها مسيق بطاقة الشبكة لنفسه في حال كانت مكتوبة كمسيق خاص للنظام Windows NT. لكن باستدعاء روتينات NDIS عوضا عن ذلك، تتحرك مسيقات NDIS المكتوبة للنظام Windows NT بسهولة إلى محيط مسيق الجهاز الظاهري في Windows.

4-9 محيط التطبيق الموزّع

لقد عرّف Andrew Tanenbaum نظاماً موزّعاً، بعكس تعريفه لشبكة حاسوب، على أنه نظام حيث «يكون وجود حواسيب مستقلة متعددة شفاف (غير مرئي) للمستعمل». أيْ يتحكم نظام تشغيل واحد بعدة حواسيب مشبّكة ويجدول معالجاتها. إن النظام Windows NT ليس نظام تشغيل موزّع. فهو يشتغل على حواسيب متعددة المعالجات ويجدول كل المعالجات لكنه يطلب من المعالجات مشاركة الذاكرة.

ورغم عدم كونه نظاماً موزّعاً، يوفّر النظام windows NT الوسائل لإنشاء تطبيقات موزّعة وتشغيلها. تستعمل المعالجة الموزّعة للإشارة إلى إمكانية مستعمل الطباعة من حاسوب واحد من عدة حواسيب عن طريق إرسال وظيفة طباعة إلى ملقّم طباعة عن بعد. وبشكل مشابه، لم يكن من غير المعتاد استعمال حاسوب بأكمله كمكان لتخزين الملفات المشاركة والتي يستطيـع المستعمل استردادها ونسخها إلى ماكنات محليّة للمعالجة. لكن حاليا، فالمعالجة الموزّعة أكثر

تعقيداً. فعوضاً عن تخزين ملفات قاعدة البيانات الكبيرة على ماكنة عن بعد ونسخها إلى ماكنة محليّة في كل مرة يريد المستعمل استعلام قاعدة البيانات، تتيح التطبيقات مثل Microsoft SQL Server للمستعمل تقديم طلب استعلام ممتلىء بعمليات البحث والفرز على الماكنة عن بعد. وعند إتمام المعالجة، ترجع فقط النتائج إلى ماكنة المستعمل يخفّض هذا النوع من احتساب المستضاف / الملقّم الحمل على قسم من النظام الذي يتصف بالقدرة الأصغر على مناولته ـ الشبكة ـ وينقل الحمل إلى معالج عن بعد، حيث يترك المعالج المحلّي حراً. إن ميزة هذه التطبيقات هي أنها توسّع قدرة الاحتساب لمحطة عمل مستعمل واحد عن طريق استخدام دورات المعالج للحواسيب الأكثر قوة عادة عن بعد.

إن هذا النوع من الاحتساب هو توسيع لنموذج المستضاف / الملقّم الذي سيق وعرض في الفصل الخامس، حيث ترسل معالجة مستضاف طلباً إلى معالجة ملقّم للتنفيذ. أما الفرق هنا فهو أن معالجة تشتغل على حاسوب مختلف. وفي نموذج المستضاف / الملقّم المحلي في Windows NT، يستعمل المعالجات وسيلة تمرير الرسائل تسمى استدعاء إجراء محلي (LPC) للإتصال عبر فسحات عنوانها. وللمعالجة الموزّعة، يحتاج إلى وسيلة تمرير رسائل شاملة. ويجب إزالة الافتراضات المتعلقة بأية معالجة سترسل الرسالة وأي حاسوب تشتغل عليه المعالجة من الوسيلة. كذلك، وبسبب عدم مشاركة معالجات المستضاف والملقّم للذاكرة (ما لم تكن تشتغل على نفس الحاسوب)، يجب أن تفترض الوسيلة أن كل البيانات ستنسخ من فسحة عنوان سري واحدة إلى أخرى على شبكة.

يمثّل احتساب المستضاف/ الملقّم طريقة تطبيق (عوضا عن نظام تشغيل) للمعالجة الموزّعة لكنه لن يستطيع النجاح دون دعم نظام التشغيل المناسب. يجب أن يزوّد نظام التشغيل ما يلي ليستخدم بنجاح احتساب المستضاف/الملقّم المشبّك:

- طريقة لإنشاء تشغيل أجزاء من تطبيق على الحواسيب المحلّية وعن بعد
- آليات بمستوى التطبيق لتمرير المعلومات بين المعالجات المحلّية وعن بعد
- الدعم لعمليات الشبكة، بما في ذلك وسائل النقل.

لقد كرّس معظم هذا الفصل لوصف القدرة الثالثة. وتعالج الأقسام الفرعية التالية أول قدرتين.

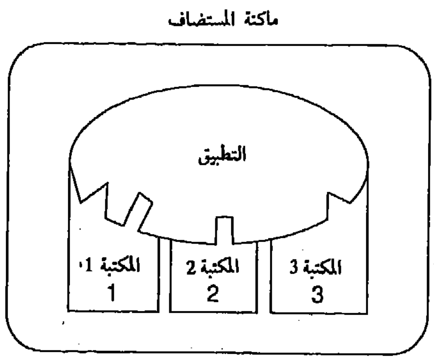
9-4-1 استدعاء الإجراء عن بعد

وسيلة استدعاء الاجراء عن بعد (RPC) هي وسيلة تتيح للمبرمج إنشاء تطبيق يتألف من أي عدد من الاجراءات، البعض منها ينفّذ محلياً والآخر ينفّذ على حواسيب عن بعد عبر

شبكة. وهي توفر معاينة إجرائية للعمليات المشبّكة عوضا عن معاينة نقلية، حيث تبسّط تطوير التطبيقات الموزّعة.

تبنى عادة براجميات تشبيك الحواسيب حول نموذج دخل / خرج لمعالجة. فمثلاً، في Windows NT، تحفز عملية شبكة عندما يصدر تطبيق طلب دخل / خرج عن بعد. ويعالجه نظام التشغيل عن طريق تقديمه إلى مغيّر وجهة الذي يعمل كنظام ملفات عن بعد. وبعد أن يستجيب النظام عن بعد للطلب ويرجع النتائج، تقاطع بطاقة الشبكة المحلّية. وتتناول النواة المقاطعة وتتم عملية الدخل / الخرج الأصلية حيث ترجع النتائج إلى المستدعي.

تستعمل الوسيلة RPC طريقة مختلفة كلياً. فتطبيقات RPC كسائر التطبيقات البنيوية الأخرى ذات برنامج رئيسي يستدعي الإجراءات أو مكتبات الإجراءات لتنفيذ المهام المحددة، كما يوضح ذلك الشكل (9-15).



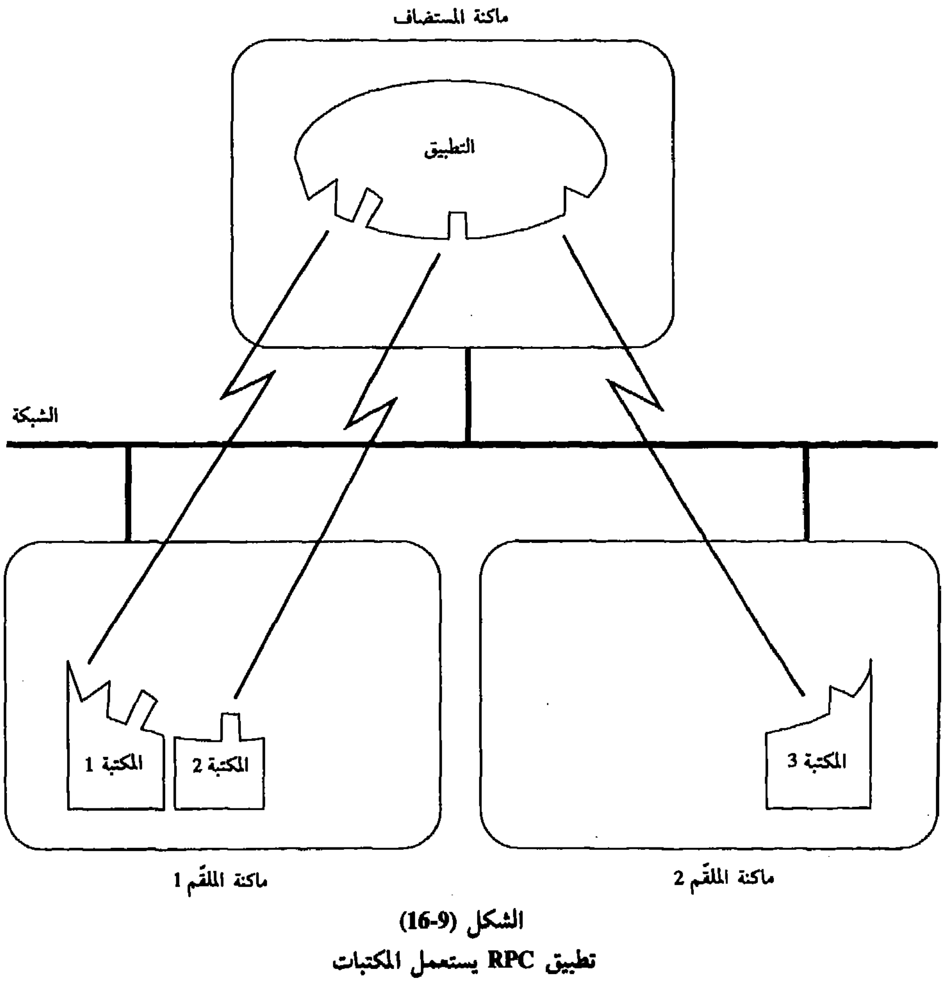
الشكل (9-15)
تطبيق يستعمل المكتبات

لكن الفرق بين تطبيقات RPC والتطبيقات العادية هو في أن بعض مكتبات الإجراءات في تطبيق RPC تنفّذ على حواسيب عن بعد بينما تنفّذ الأخرى محلياً، كما يظهر ذلك الشكل (9-16).

وبالنسبة لتطبيق RPC تبدو كل الاجراءات وكأنها تنفّذ محليّا. بمعنى آخر، عوضاً عن جعل المبرمج يكتب شيفرة لإرسال الطلبات الحسابية أو المتعلقة بالدخل / الخرج عبر شبكة ومناولة بروتوكولات الشبكة ومعالجة أخطاء الشبكة وانتظار النتائج وما شابه، تتناول براجميات RPC هذه المهام تلقائياً. وتستطيع وسيلة RPC في Windows NT العمل على أية آليات نقل متوفرة محمّلة في النظام.

لكتابة تطبيق RPC، يقرر المبرمج الإجراءات التي ستنفّذ محلياً وتلك التي ستنفّذ عن بعد. فمثلاً، إفترض، أن محطة عمل عادية تحتوي وصلة شبكة إلى حاسوب فائق Cray أو إلى ماكنة مصممة خصيصاً للعمليات المتجهية العالية السرعة. وفي حال كتب المبرمج تطبيقاً يعالج مصفوفات كبيرة، فمن المنطقي إرسال العمليات الحسابية إلى حاسوب عن بعد عن طريق كتابة البرنامج كتطبيق RPC.

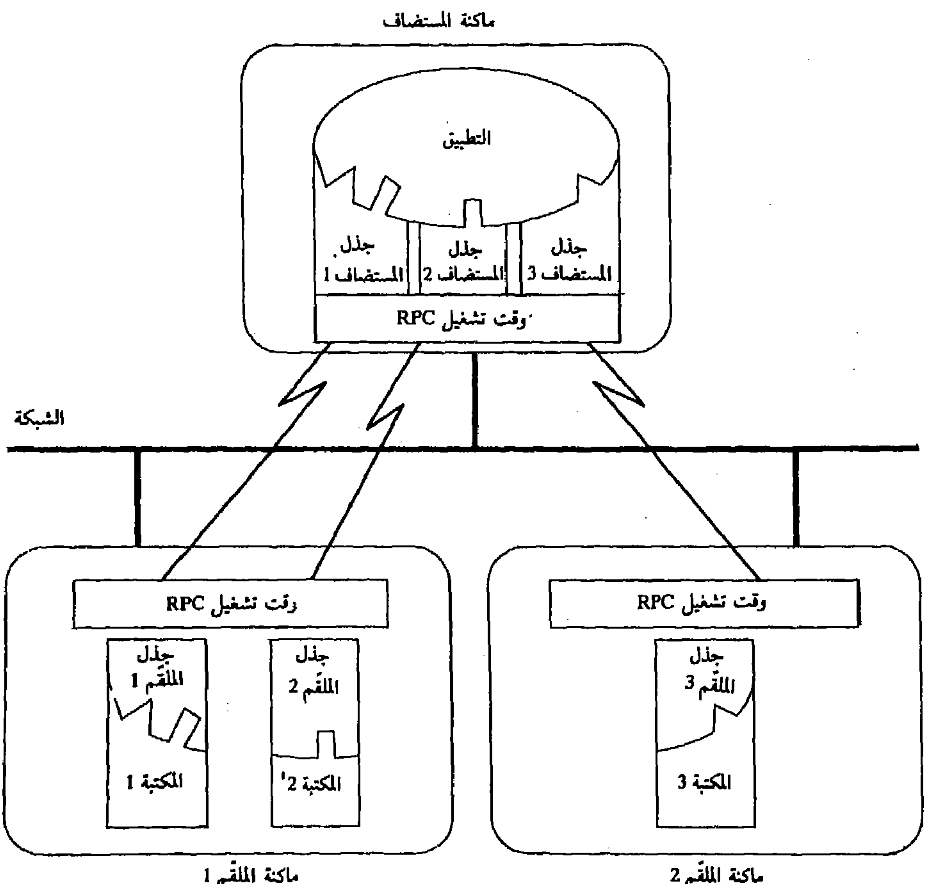
تعمل تطبيقات RPC بهذه الطريقة: عند اشتغال تطبيق، فإنه يستدعي إجراءات محلّية وأيضاً إجراءات غير موجودة على الماكنة المحلّية. ولمناولة هذه الحالة الأخيرة، يربط التطبيق إلى مكتبة DLL محليّة تحتوي إجراءات جذليّة، واحدة لكل إجراء عن بعد. تتصف الإجراءات



الشكل (9-16)
تطبيق RPC يستعمل المكتبات

الجذلية بنفس الاسم وتستعمل نفس تداخل الإجراءات عن بعد، لكن وعوضاً عن تنفيذ العمليات المطلوبة، يستلم الجذل البارامترات الممرّة إليه وينظّمها للإرسال عبر الشبكة. ويعين تنظيم البارامترات ترتيبها وحزمها بطريقة معيّنة لتناسب رابط شبكة مثل التدقيق في المراجع وانتقاء نسخة عن أية بنيّة بيانات يشير إليها مؤشر.

بعد ذلك، يستدعي الجذل إجراءات وقت تشغيل RPC التي تحدد موقع الحاسوب حيث يستقر الإجراء عن بعد، ويحدد آليات النقل التي يستعملها الحاسوب ويرسل الطلب إليها براجيات النقل المحلّية. وعندما يستلم الملقّم عن بعد طلب RPC فإنه يلغي تنظيم البارامترات (عكس تنظيمها) ويعيد إنشاء استدعاء الإجراء الأصلي ويستدعي الإجراء. وعند انتهاء الملقّم،



الشكل (9-17)

وقت تشغيل RPC

ينفّذ الملقّم تتابع عكسي لإرجاع النتائج إلى المستدعي. يوضح وقت تشغيل RPC في الشكل (17-9).

إضافة إلى وقت تشغيل RPC، تشمل وسيلة RPC في Microsoft مصرّف يسمى مصرّف لغة تعريف التداخل في Microsoft (MIDL) إنشاء تطبيق RPC. يكتب المبرمج من نماذج الوظيفة الأوليّة (بافتراض تطبيق C أو C++) يصف الروتينات عن بعد ثم يضع الروتينات في ملف. بعد ذلك، يضيف المبرمج إلى هذه النماذج الأوليّة بعض المعلومات الإضافية، مثل معرّف خاص بالشبكة لحزمة الروتينات ورقم الإصدار، إضافة إلى الصفات التي تحدد أن البارامترات هي بارامترات دخل، خرج أو كلاهما. وتشكّل النماذج الأوليّة المزخرفة ملف لغة تعريف التداخل للمطوّر (IDL).

بعد إنشاء الملف IDL، يصرّفه المبرمج مع المصرّف MIDL الأمر الذي يؤدي إلى روتينات جذلية لجهة المستضاف ولجهة الملقّم وكذلك ملفات رأسية يجب شملها في التطبيق. وعند ربط تطبيق جهة المستضاف بملف الروتينات الجذلية، يدقق بكل مراجع الإجراءات عن بعد. بعد ذلك، تركّب الإجراءات عن بعد باستعمال معالجة مشابهة على ماكنة الملقّم. ويحتاج المبرمج الذي يرغب باستدعاء تطبيق RPC موجود إلى كتابة جهة المستضاف فقط من البراجيات وربط التطبيق إلى وسيلة وقت تشغيل RPC المحليّة.

يستعمل وقت تشغيل RPC تداخلاً موفّر النقل RPC شاملاً للتخاطب مع بروتوكول نقل. ويعمل تداخل الموفّر كطبقة رقيقة بين الوسيلة RPC والنقل حيث يخطط عمليات RPC على الوظائف المتوفرة من قبل آلية النقل. تستخدم الوسيلة RPC في Windows NT مكتبات DLL في موفّر النقل للأنابيب المسمّاة، NetBIOS و TCP/IP و DECnet. ويمكن دعم آليات نقل إضافية عن طريق كتابة مكتبات DLL موفّر جديدة. وبطريقة مشابهة، صممت الوسيلة RPC للعمل مع وسائل الآمان للشبكة المختلفة. وكمكتبات DLL لموفّر النقل، يمكن إضافة مكتبات DLL للأمان بين الوسيلة RPC والشبكة. وفي غياب مكتبات DLL للأمان الأخرى، تستعمل براجيات RPC في Windows NT الأمان الداخلي للأنابيب المسماة. (يصف القسم 9-4-2 الأنابيب المسماة بتفصيل أكبر).

ولكي تعمل داخلياً وسيلة RPC واحدة مع تطبيقات RPC على الماكنات الأخرى، يجب على كلاهما إستعمال نفس مصطلحات RPC. تتوافق الوسيلة RPC من /خلافحسحبغ مع المواصفات القياسيّة RPC المعرّفة من قِبَل مؤسسة البراجيّات المنفتحة (OSF) في مواصفات محيط الإحتساب الموزّع (DCE) الخاصة بها. وهكذا، تستطيع التطبيقات المكتوبة بإستعمال الوسيلة

RPC من Microsoft إستدعاء إجراءات عن بعد المتوفّرة على الأنظمة الأخرى التي تستعمل المواصفات القياسيّة DCE.

إن معظم خدمات تشبيك الحواسيب في Windows NT هي تطبيقات RPC وهذا يعني أنه يمكن إستدعاءها من قبَل المعالجات المحليّة ومن قِبل المعالجات على الحواسيب عن بُعد. وبالتالي، يستطيع حاسوب مستضاف عن بُعد إستدعاء الملقّم أو يمكنه إستدعاء خدمة المرسل لتوجيه الرسائل إليك (وكل هذه عرضة لتقييدات الأمان طبعاً). لقد إعتبر Chuck Lenzmeier، مطوّر الملقّم في Windows NT، خدمات الممكّنة في RPC على أنها الأجدر وأكثر المزايا إفادة في تشبيك الحواسيب في Windows NT.

2-4-9 الأنابيب المسمّاة:

لقد إعتبرت الأنابيب المسمّاة أصلاً من قبَل Microsoft كتداخل عالي المستوى مع NetBIOS. وقد وفّر NetBIOS لتطبيقات تشبيك الحواسيب ما وفّره BIOS للنظام MS-DOS ــ فقد جرّدت العتاد. وهكذا، إعتمدت معاينة بمستوى منخفض لإتصالات الشبكة. توفّر الأنابيب المسمّاة تداخلاً أكثر تجريداً (ومريحاً) إلى الشبكة. وعوَضاً عن الإهتمام بالتسليك، وإرسال البيانات وما شابه، الذي يستطيع مبرمج يستعمل الأنابيب المسمّاة، فتح إنبوب ووضع البيانات فيه. ويقوم مستعمل الأنبوب بفتحه وقراءة البيانات منه. وتتمّ مناولة تسليم الحاسوب التقاطعي تلقائياً، ويعادل إستدعاء أنبوب مسمّى واحد العديد من العمليّات بمستوى النقل.

تستخدم الأنابيب المسمّاة في Windows NT بواسطة مسيّق نظام ملفّات الأنابيب المسمّاة، وهو نظام ملفّات إنتقاليّة يخزّن بيانات الأنبوب في الذاكرة ويستردّها عند الطلب. وهو يعمل كنظام ملفّات عادي عند معالجة طلبات أنبوب مسمّى محلياً أو عند إسترداد طلب أنبوب مسمّى من حاسوب عن بُعد.

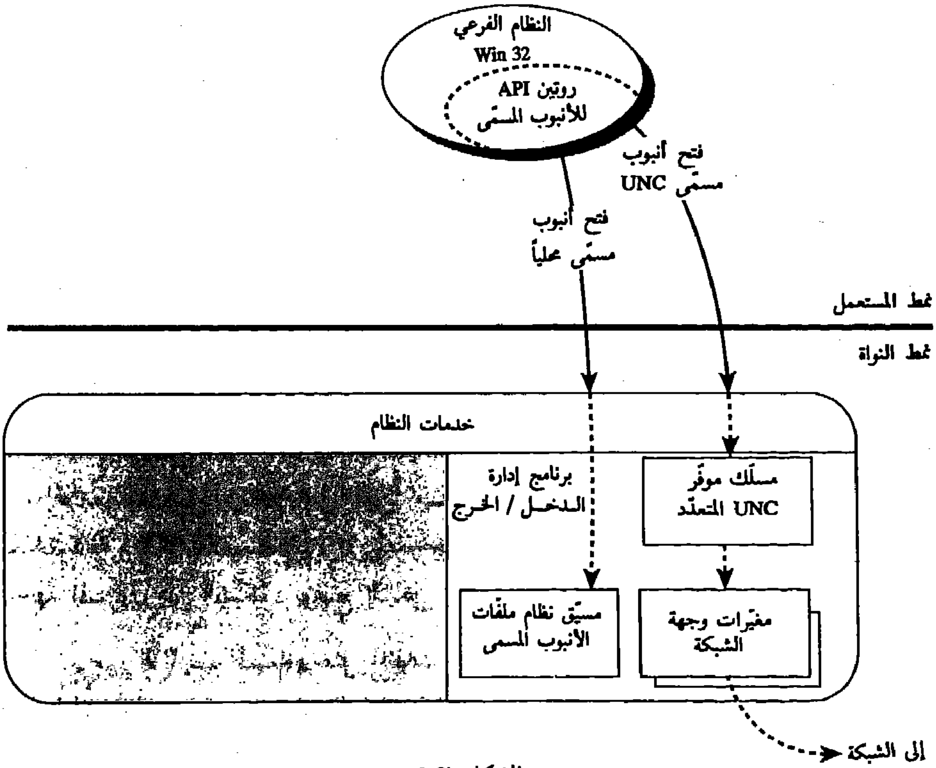
عندما يستعمل برنامج محلي أنبوباً مسمى يحتوي على إسم UNC (شبكة)، تتواصل المعالجة كأية معالجة لطلب «ملفّ» عن بُعد آخر. يعترض مسيّق MUP الطلب ويرسله إلى مغيّر الوجهة المسؤول عن الشبكة. يوضح الشكل (9-18) معالجة الأنبوب المسمى المحلي وعن بُعد.

تعرض الأنابيب المسمّاة، مثل الملفّات، ككائنات ملفّ في Windows NT وتعمل بظلّ نفس آليات أمان كائنات البرنامج التنفيذي NT الأخرى. وعندما تحاول شعبة محليّة فتح أنبوب مسمّى، يدقّق الوصول المطلوب من قبَل الشعبة مقابل اللائحة ACL على كائن ملف الأنبوب المسمّى. فإذا لم يحصل أي تطابق، يمنع الوصول. وهكذا، تحتوي وسيلة الأنبوب المسمّى على

أمان داخلي. إضافة لذلك، فإنها تتيح لمعالجة واحدة إعتماد سياق الأمان العائد لمعالجة أخرى، وهذه قدرة تسمى التقليد. تتيح هذه القدرة لنظام فرعي إستعمال هويّة معالجة مستضاف عند فتح أنبوب مسمّى عن بُعد، على سبيل المثال.

ولأنه يمكن أن تتواجد على الحواسيب المحليّة والبعيدة، توفّر الأنابيب المسمّاة وسيلة لمعالجات المستضاف والملقّم في تطبيق موزّع من الإتصال ومشاركة البيانات. تخفي وسيلة الإتصال بالمعالجة الداخليّة الإتصال بين الماكنات الداخلية عن التطبيق. وتحت روتينات API العائدة لها، تستعمل وسيلة الأنبوب المسمّى إحدى آليات النقل بمستوى منخفض لإرسال بياناتها. وبعكس RPC، تعمل وسيلة الأنبوب المسمّى على نموذج يعتمد على الدخل / الخرج وهي مفيدة لإرسال دفق البيانات من معالجة واحدة إلى أخرى.

تستطيع الوسيلة RPC الإشتغال على آليات نقل متعدّدة بإستعمال أنواع مختلفة من آليات الأمان. لكن، عند العمل على شبكة LAN Manager، تستعمل وسيلة RPC الأنابيب المسمّاة



الشكل (9-18)
معالجة الأنبوب المسمّى لجهة المستضاف

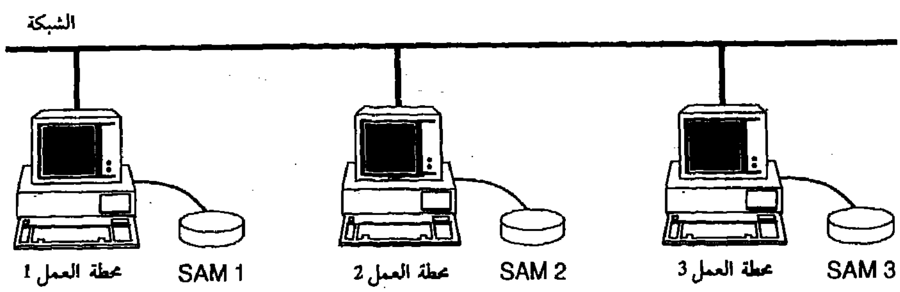
كآلية نقل الشبكة الخاصة بها. وعن طريق القيام بذلك، تستفيد عمليات RPC من الأمان الداخلي الذي يطبّقه النظام Windows NT على الأنابيب المسمّاة.

5-9 تشبيك حواسيب الشركات ونظام الأمان الموزّع:

يزوّد نظام Windows NT القياسي مع قدرات ملقّم داخليّة. يمكّن الملقّم عمليّات مجموعة العمل، مثل نسخ الملفّات بين النظامين أو إعداد طابعة يمكن أن تشارك من قبل عدّة محطات عمل. هذا النوع من تشبيك الحواسيب الصغيرة النطاق مفيدة للمكاتب الصغيرة وشبكات المنازل أو محطات العمل الإفراديّة الموصولة إلى شبكات بواسطة الخطوط الهاتفيّة. لكن في المكاتب الأكبر أو المختبرات، قد يحتاج لوسائل إضافيّة.

لقد شرحت الفصول السابقة من هذا الكتاب أوجه الأمان المختلفة في النظام Windows NT. والأمن جزء هام من أية عمليّة شبكة وضروري لحماية بيانات مستعمل واحد من الوصول إليها من قبل المستعملين الآخرين ومن العبث في سجلات الشركة من قبَل الدخلاء وما شابه. لكن نظام الأمان يحتوي على بعض الأمور الإداريّة المتعلّقة به. فمثلاً، يصف الفصل الخامس، «Windows والأنظمة الفرعيّة المحميّة»، كيف يتوجّب على المستعمل التسجيل على النظام Windows NT وكيف يتمّ التحقّق من المعلومات المسجّلة من قبل نظام التشغيل. ولكي يتعرّف نظام التشغيل إلى المستعمل، يجب أن يعدّ المدير حساباً للمستعمل على النظام الذي يريد الوصول إليه. يخزّن النظام Windows NT الأسماء وكلمات السرّ في قاعدة بيانات تسمّى قاعدة بيانات إدارة حسابات الأمان (SAM).

عندما يحاول مستعمل التسجيل على محطة العمل، يدقّق النظام Windows NT بقاعدة البيانات SAM المتعلّقة بمحطة العمل العائدة له ليتحقّق في كلمة السرّ. وفي شبكة الأنظمة

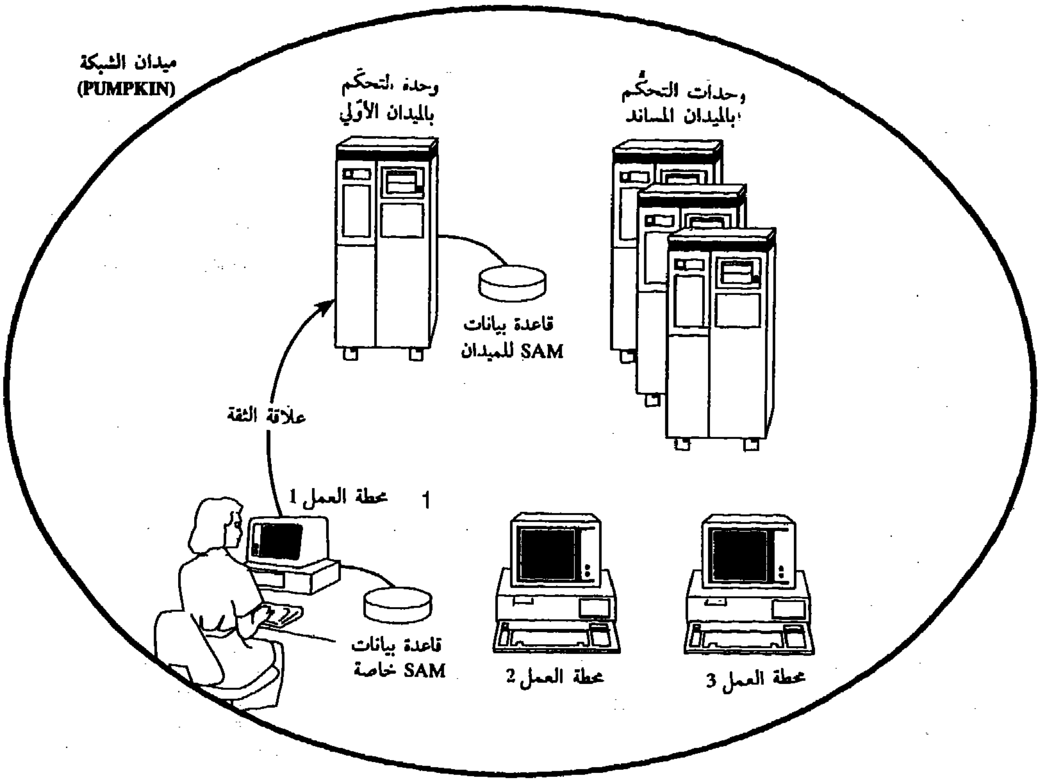


الشكل (9-19)
تشكيل الشبكة الصغيرة

الصغيرة، يشغّل كل مستعمل محطة عمل Windows NT وكل محطة عمل تحتوي على قاعدة بيانات خاصة لحسابات المستعمل، كما يوضح الشكل (9-19).

إذا أراد مستعمل معيّن الوصول إلى ملفّات (أو أجهزة) على كل من محطات العمل، يجب أن يحتوي على حساب مستقلّ على كل ماكنة. وهذا يعني أنه إذا أراد أن يغيّر كلمة السرّ وأراد الإحتفاظ بنفس كلمة السرّ على كل ماكنة يحاول الوصول إليها، يجب عليه أن يحدّث كلمة السرّ بشكل مستقلّ على كل نظام.

هذا النوع من صيانة النظام مزعج لإنشاءات الشبكة في المؤسّسات التجاريّة الكبيرة. فعندما يزداد عدد محطات العمل، تزداد مهمة إدارتها تناسبياً. ولإستيعاب حاجات شبكات المؤسّسات التجاريّة، أدخل Cliff Van Dyke و Jim Kelly و Jim Horne تحسينات على براجيّات الشبكة الداخليّة في Windows NT. وقد سمّي Lan Manager للنظام Windows NT، وقد



الشكل (9-20)
تشكيل الشبكة ذات الحجم المتوسط

أضافت هذه البراجيّات القدرات لدعم تشبيك الحواسيب الند للند الموجودة في محطة عمل Windows NT وهي تتيح إنشاء ميدان شبكة (يظهر إصدار أولي في LAN Manager 2.x) لتبسيط مهمّة إدارة النظام. يوضح الشكل (9-20) ميدان شبكة عيّنة.

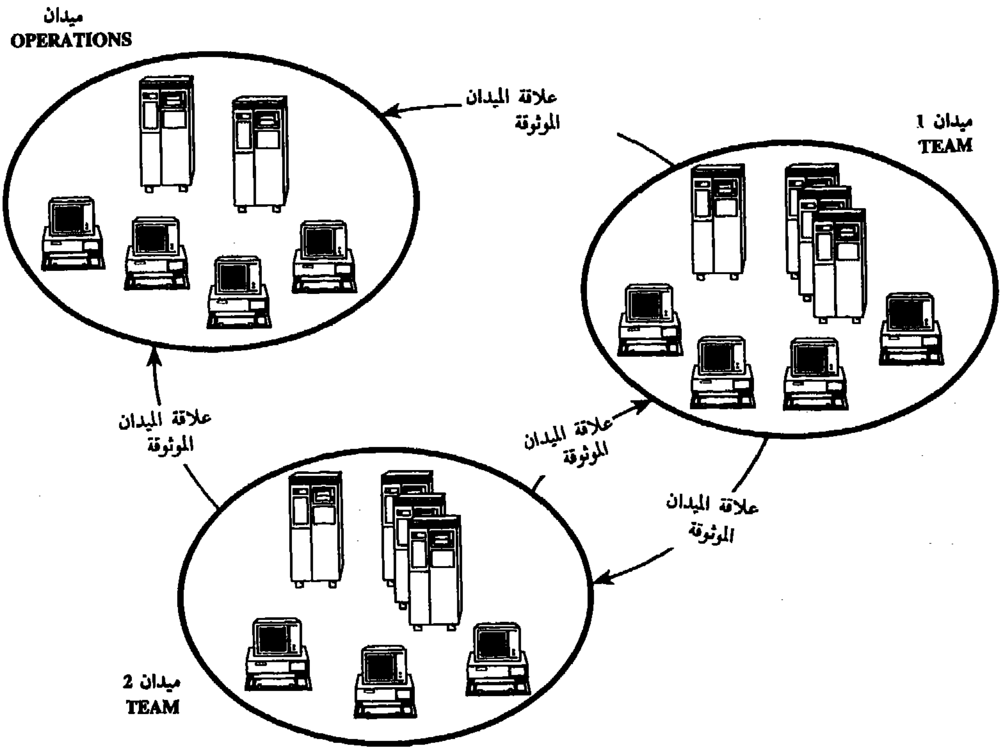
في هذا الشكل، تمثّل الدائرة الكبيرة ميدان شبكة حيث تربط كل الحواسيب. يتضمّن الميدان محطات العمل وعدّة ماكنات ملقّم، وتسمّى هذه الأخيرة وحدات التحكّم بالميدان. عندما يسجّل مستعمل نفسه، فإنه ينتقي التسجيل على حساب معرّف في محطة العمل الخاص به أو على حساب موجود في الميدان الأوّلي الخاص به، وهو الميدان الذي تعود إليه الماكنة.

إذا سجّل المستعمل على حساب على محطة العمل الخاصة به، تستعمل براجيّات التحقيق بالأصالة المحليّة المعلومات المخزّنة في قاعدة بيانات SAM لمحطة العمل للتحقّق من أصالة التسجيل. من الناحية المقابلة، إذا سجّل المستعمل على الميدان، ترسل براجيّات التحقّق من الأصالة المحليّة طلب التسجيل إلى الميدان للتحقّق منه. تحتوي وحدة التحكّم بالميدان الأوّلي على قاعدة بيانات SAM تطبق على كامل الميدان، وتحافظ وحدات التحكّم بالميدان المساند على نسخ قاعدة البيانات. يحرّر هذا الأمر المستعمل من الحصول على الحسابات على كلّ ملقّم ويحسّن تفاوت الأخطاء. وإذا تعطّلت وحدة تحكّم بميدان معيّن، يستطيع النظام ديناميّاً توجيه طلب تسجيل إلى ملقّم مختلف.

في الشكل (9-20)، يمثّل الخط بين محطة العمل ووحدة التحكّم بالميدان الأوّلي علاقة ثقة، وهو بند أمان يعني أن محطة العمل «تثق» بالميدان لجهة تحديد أصالة تسجيل المستعمل. تتيح علاقة الثقة للنظام Windows NT إعداد قناة آمنة بين النظامين والوصول إلى الموارد على الميدان.

أما التأثير الجانبي لإنشاء ميدان فهو في إمكانيّة تسجيل المستعمل على الميدان من أية محطة عمل أخرى (أو ملقّم) في الميدان والوصول إلى محطة العمل الخاصة به عن بُعد. فمثلاً، إذا أنشأت مجموعة مشروع تطوير أو مختبر إختبار لأنظمة Windows NT، وتمّ تجميع هذه الأنظمة في ميدان، يستطيع مستعمل عبقري على حساب في الميدان من التسجيل على حسابه من أي نظام في المختبر.

إن القدرة على إنشاء ميدان شبكة مناسبة لإنشاءات الشبكة من الحجم المتوسّط حيث يستعمل عدّة ملقّمات من قبَل عدد كبير من محطات العمل وفي الإنشاءات الأكبر، مثل شبكات المؤسّسات التجاريّة الكبيرة، تصبح الميادين أكثر قيمة. فهي تتيح للشركة تقسيم مواردها إلى عدّة وحدات سريّة (ميدان) وإدارة هذه الوحدات بطريقة مرنة. يظهر الشكل (9-21) تشك شبكة كبيرة.



الشكل (9-21)
تشكيل الشبكة الكبيرة

تحتوي هذه الشبكة ثلاثة ميادين مختلفة: ميدانان لمجموعات التطوير (يسمّيان TEAM 1 و TEAM 2) وواحد لموظفي إدارة النظام (يسمى OPERATIONS). تتواجد علاقات الميدان الموثوقة بين ميداني التطوير وبين ميداني التطوير وميدان العمليّات. تتيح هذه البنية لمطوّر من المجموعة TEAM 1 التسجيل على ميدانه الأوّلي من ماكنة في ميدان المجموعة TEAM 2، على سبيل المثال. لكن المميّز في ذلك أن هذه البنية تتيح لفرد من مجموعة العمليّات التسجيل على حسابه والوصول إلى الموارد في ميداني المجموعتين TEAM 1 و TEAM 2 إذا كانا عائدين لميدان OPERATIONS. لقد تطلّبت الإصدارات السابقة من LAN Manager من موظفي العمليات الحصول على حساب على كل ميدان مطلوب إدارته، لكن إصدار LAN Manager للنظام Windows NT يزيل هذا الطلب. في هذا المثال، وبعد أن يسجّل مدير نظام، فإنه يستطيع الوصول إلى الموارد على الميادين الأخرى. وعند التوصيل إلى الميدان TEAM 1، تدقّق براجيّات الأمان في TEAM 1 لجهة وجود علاقة ثقة مع الميدان OPERATIONS. وبما أن علاقة الثقة

موجودة يستعمل الميدان TEAM 1 قناة الأمن العائدة له لتمرير طلب التسجيل إلى الميدان OPERATIONS، الذي يتحقّق من أصالة المستعمل للوصول إلى قاعدة بيانات SAM للميدان OPERATIONS. وإذا كانت الأصالة صحيحة، يستطيع مدير النظام تركيب البراجيّات وتنفيذ عمليات المساندة وإتمام أعمال الصيانة الأخرى. يتيح هذا النوع من الأمان الموزّع حتى لمؤسّسات كبيرة ذات ميادين متعدّدة من إدارة الموارد بسهولة مع توفير الوصول إلى كل موارد الشبكة من أي مكان في الشبكة.

رغم أن تشكيلات الشبكة المتوسطة والكبيرة تتيح للمستعمل المرونة في الوصول إلى الموارد، يستطيع كل مستعمل إعداد ماكنته لتحديد وصول الآخرين إليه بالطرق التالية:

- عن طريق تطبيق لوائح التحكُّم بالوصول (ACL) على الملفّات والموارد المحليّة الأخرى التي تسمح أو تمنع الوصول إلى المستعملين الإفراديين أو إلى مجموعات المستعملين.
- عن طريق تعيين (أو عدم تعيين) تفضيلات إلى المستعملين الإفراديين أو مجموعات المستعملين.
- عن طريق الإتاحة الواضحة للأفراد أو المجموعات من التسجيل بطرق محدّدة، بإستعمال إحدى هذه الوسائل أو أكثر:
  - التفاعلي. التسجيل من لوحة المفاتيح.
  - الشبكة. التسجيل عبر وصلة شبكة.
  - الخدمات. التسجيل كخدمة (مثل خدمة المرسل أو خدمة المحذّر).

عن طريق منع التفضيلات عن فرد أو مجموعة من الأفراد، وعن طريق تعيين لوائح ACL إلى الكائنات المحليّة، وعن طريق تحديد المستعمل الذي يستطيع التسجيل على محطة عمل إفرادية، يواصل مستعمل التحكُّم بمحيطه. وإذا رغب المستعمل، يستطيع منع أي شخص بإستثناء نفسه من الوصول إلى ماكنته لأي سبب. ونتيجة تعدُّد الإستعمالات، يشير المطوِّرون إلى براجيّات إدارة النظام Windows NT بالإسم «FlexAdmin».

إضافة إلى الميادين، والميادين الموثوقة والأمان الموزّع، يوفّر LAN Manager لبراجيّات NT أدوات إنشاء نسخات مرآتيّة وتشريط القرص وإستنساخ الملفّات وإدارة الملقّم التخطيطي.

9-6 بإختصار:

لقد أدّى تكاثر شبكات الحواسيب الكبيرة وحاجة المستعملين للإتصال ومشاركة قواعد البيانات المركزيّة إلى رفع براجيّات تشبيك الحواسيب من عالم الإفادة إلى عالم الحاجة. وعن

طريق المشاركة في شبكة واحدة أو أكثر، يستطيع نظام تشغيل زيادة طاقة المعالجة وقدرات التخزين والإتاحة للمستعمل الإتصال ومشاركة البيانات وتوفير مجموعة غنيّة من القدرات لتطبيقاته مقارنة مع ما يوفّره لوحده. لتوفير هذه الميزات بطريقة فعّالة، تركب داخلياً براجيّات تشبيك الحواسيب في Windows NT، حيث تعمل على أساس متساوي مع بقيّة البرنامج التنفيذي NT. ورغم كونها داخليّة، لا تُسلّك بأسلاك مكوّنات الشبكة في النظام. تتيح نماذج مكتبة DLL ونظام الدخل / الخرج المرنة في Windows NT إضافة براجيّات الشبكة إلى نظام التشغيل وإزالتها منه ديناميّاً.

يوفّر النظام Windows NT عدّة تداخلات شبكيّة تتيح لها الربط مع الأنواع المختلفة من الشبكات ومن التفاعل مع الأنواع المختلفة من أنظمة الحواسيب. ويتيح تداخل الموفّر لبائعي الشبكات غير Microsoft الربط مع روتينات API للدخل / الخرج للملفّات والتصفّح في Win 32. وتتيح طبقة TDI لمغيّرات الوجهة وللمقّمات الشبكة من الوصول وإستعمال أي مسيّق نقل متوفّر دون تعديل شيفرتها. وتتيح طبقة NDIS لمسيّقات النقل الوصول إلى أية بطاقة شبكة وتوفّر لمسيّقات النقليّة من أنظمة Windows NT إلى محيط مسيّق الجهاز الظاهري في Windows.

تتيح وسائل التطبيق الموزّع بما في ذلك روتينات RPC وآليات الإتصال بين المعالجات لمطوّري التطبيقات إستخدام الحواسيب المشبّكة عن طريق إلغاء تحميل الأعمال المكثّفة الحسابيّة إلى الماكنات الأخرى أو عن طريق الوصول إلى الموارد عن بعد إذا كانت محليّة. إضافة لذلك، يوسّع إصدار LAN Manager للنظام Windows NT قدرات تشبيك الحواسيب لنظام Windows NT القياسي لاستيعاب شبكات المؤسّسات التجاريّة الكبيرة مع وسائل الأمان الموزّع والإدارة. ومع قدرات تشبيك الحواسيب المكثّفة، يستطيع النظام Windows NT تحويل محطة عمل لسطح مكتب عادي إلى شبكة مكلفة من موارد الإحتساب.

معجم الاصطلاحات والمختصرات

access control entry (ACE) ـ إدخال التحكُّم بالوصول. إدخال في قائمة التحكُّم بالوصول (ACL). وهو يحتوي بطاقة تعريف أمان (SID) ومجموعة من حقوق الوصول. ويُتاح لمعالجة ذات بطاقة SID مطابقة حقوق الوصول المسردة ومنعها أو إتاحتها مع تدقيق. راجع أيضاً قائمة التحكُّم بالوصول.

access control List (ACL) ـ قائمة التحكُّم بالوصول. قسم من واصف الأمان الذي يسرد الحماية المسلّطة على كائن. يحتوي مالك الكائن على تحكُّم بالوصول إستنسابي للكائن ويستطيع تغيير قائمة ACL للكائن للإتاحة للآخرين الوصول إلى الكائن أو عدم إتاحة ذلك. تتألّف قائمات التحكُّم بالوصول من إدخالات تحكُّم بالوصول (ACEs). راجع أيضاً إدخال التحكُّم بالوصول، والتحكُّم بالوصول الإستنسابي، وواصف الأمان.

access right ـ حقوق الوصول. إذن ممنوح لمعالجة لمناولة كائن معيّن بطريقة معيّنة (مثلاً، عن طريق إستدعاء خدمة). تدعم أنواع كائنات مختلفة حقوق وصول مختلفة، مخزّنة في قائمة التحكُّم بالوصول لكائن (ACL). راجع أيضاً إدخال التحكُّم بالوصول وقائمة التحكُّم بالوصول.

access token ـ تأشيرة الوصول. كائن يعرّف بشكل خاص مستعمل تسجّل للتوّ. تثبّت كل تأشيرة وصول إلى كل معالجات المستعمل وتحتوي بطاقة تعريف الأمان (SID) للمستعمل وأسماء أية مجموعة ينتمي إليها المستعمل وأية تفضيلات يمتلكها المستعمل والمالك المفترض لأي كائنات أنشأتها معالجات المستعمل وقائمة التحكُّم بالوصول (ACL) المفترضة الواجب تطبيقها على أي كائن تنشئه معالجات المستعمل. راجع أيضاً بطاقة تعريف الأمان.

ACE ـ إدخال التحكُّم بالوصول.

ACL ـ قائمة التحكُّم بالوصول.

address space ـ فسحة العنوان. راجع فسحة العنوان الظاهري.

alert ـ تنبيه. إبلاغ غير متزامن عن إرسال شعبة واحدة إلى أخرى. يقاطع التنبيه الشعبة المستلمة عند نقاط معرّفة بشكل واضح في تنفيذها وتجعلها تنفّذ إستدعاء إجراء غير متزامن (APC). راجع أيضاً الشعبة المنبّهة وإستدعاء الإجراء غير المتزامن.

alertable thread ـ الشعبة المنبّهة. شعبة أعلنت نفسها جاهزة لتنفيذ إستدعاء إجراء غير متزامن (APC). تصبح الشعبة منبّهة إما عن طريق إنتظار مقبض كائن وتحديد الإنتظار على أنه منبّه أوعن طريق الإختبار لجهة وجود إستدعاء APC معلّق. راجع أيضاً التنبيه وإستدعاء الإجراء غير المتزامن.

alerter service ـ خدمة المنبّه. خدمة شبكة ترسل رسائل النظام إلى مستعمل. راجع أيضاً الخدمة.

alerting a thread ـ تنبيه شعبة. راجع التنبيه.

APC ـ إستدعاء إجراء غير متزامن.

APC object ـ كائن إستدعاء إجراء غير متزامن. تمثيل النواة لإستدعاء إجراء غير متزامن (APC). وهو كائن تحكّم يحتوي عنوان إستدعاء APC ومؤشّر إلى كائن الشعبة الذي سينفّذه. راجع أيضاً إستدعاء الإجراء غير المتزامن وصفيفة APC.

APC queue ـ صفيفة إستدعاء الإجراء غير المتزامن. قائمة بكائنات إستدعاء الإجراء غير المتزامن (APC) الواجب تنفيذها من قبَل شعبة معيّنة. يؤدّي تواجد كائن APC في صفيفة APC لكائن إلى مقاطعة براجيّات عند مستوى طلب مقاطعة (IRQL) لإستدعاء APC عند تنفيذ الشعبة في المرّة التالية (في حال تواجدت شروط تمكين أخرى). راجع أيضاً إستدعاء الإجراء غير المتزامن وكائن APC.

API ـ تداخل البرمجة التطبيقيّة.

(API) application programming interface ـ تداخل البرمجة التطبيقيّة. مجموعة من الروتينات التي يستعملها برنامج تطبيقي لطلب وتنفيذ خدمات بمستوى منخفض المنفّذة من قبل نظام تشغيل.

ASMP ـ معالجة متعدّدة غير متماثلة.

associated IRPs ـ حزمات طلب دخل / خرج متعلّقة. مجموعة من حزمات طلب دخل /

خرج (IRPs) منشأة لمعالجة طلب دخل / خرج واحد. تؤدّي كل حزمة IRP متعلّقة إلى إستيفاء قسم معيّن من الطلب. وعند معالجة كل حزمات IRP المتعلّقة، يتمّم طلب الدخل / الخرج. راجع أيضاً حزمة طلب الدخل / الخرج.

asymmetric multiprocessing (ASMP) ـ المعالجة المتعدّدة غير المتماثلة. نظام تشغيل متعدّد المعالجة ينتقي دائماً نفس المعالج لتنفيذ شيفرة نظام تشغيل بينما تشغّل المعالجات الأخرى وظائف المستعمل فقط. راجع أيضاً المعالجة المتعدّدة والمعالجة المتعدّدة المتماثلة.

asynchronous ـ غير متزامن. يحصل في أي وقت دون إعتبار للإنسياب الرئيسي لبرنامج (مثلاً، مقاطعة جهاز). قارن مع المتزامن.

asynchronous I/O ـ الدخل / الخرج غير المتزامن. نموذج دخل / خرج حيث يصدر تطبيق طلب دخل / خرج ثم يواصل التنفيذ خلال نقل الجهاز للبيانات. يزامن التطبيق مع إتمام نقل البيانات عن طريق إنتظار مقبض ملفّ أو مقبض حدث. قارن مع الدخل / الخرج المتزامن.

asynchronous procedure call (APC) ـ إستدعاء الإجراء غير المتزامن. وظيفة تنفّذ لاتزامنياً في سياق شعبة معيّنة. تصدر النواة مقاطعة براجيّات عند تنفيذ الشعبة (إذا تواجدت شروط تمكين أخرى) وتوجّه الشعبة لتنفيذ إستدعاء APC. راجع أيضاً كائن APC وصفيفة APC.

attribute caching ـ تخبئة الصفة. طريقة تستعمل في النظام الفرعي Win 32 لتحقيق كسب أداء عندما يستدعي تطبيق Win 32 وظائف الرسم. تتذكّر مكتبة الربط الديناميّ (DLL) لجهة المستضاف عندما يغيّر تطبيق بعض صفات عرض الشاشة ويرسل البيانات إلى ملقّم Win 32 عندما يرسم التطبيق شيئاً ما على الشاشة. راجع أيضاً التخريم.

auditing ـ التدقيق. القدرة على كشف الأحداث المهمّة المتعلّقة بالأمان وتسجيلها وخاصة أية محاولة لإنشاء كائنات أو الوصول إليها أو حذفها. يستعمل نظام الأمان في Windows NT بطاقات تعريف الأمان (SIDs) لتسجيل المعالجة التي نفّذت الفعل. راجع أيضاً بطاقة تعريف الأمان.

authentication ـ التحقّق من الأصالة. صلاحية معلومات تسجيل مستعمل. تنفّذ بواسطة حزمة التحقّق من الأصالة بالتوافق مع النظام الفرعي للأمان في Windows NT. راجع أيضاً حزمة التحقّق من الأصالة.

authentication package حزمة التحقّق من الأصالة. منظومة براجيّات يمكن وصلها إلى نظام الأمان في Windows NT. للتحقّق من أصالة تسجيلات المستعمل لأجهزة الدخل المتنوّعة. راجع أيضاً حزمة التحقّق من الأصالة.

automatic working set trimming ـ التهذيب التلقائي لمجموعة العمل. طريقة مستعملة من قبَل برنامج إدارة الذاكرة الظاهريّة (VM) في NT لزيادة كميّة الذاكرة الخالية المتوفّرة في النظام. وهي تخفّض حجم مجموعة العمل للمعالجة عندما تنخفض كميّة الذاكرة الخالية. راجع أيضاً مجموعة العمل.

backing store ـ المخزن المساند. وسيط تخزين، مثل القرص، يستخدم «كذاكرة» مساندة لترتيب الصفحات عندما تمتلىء الذاكرة الفعليّة. راجع أيضاً الترتيب في صفحات.

batching ـ التوزيع. طريقة مستعملة في النظام الفرعي Win 32 لتحقيق كسب أداء عندما يستدعي تطبيق Win 32 وظائف رسم. تخزّن مكتبة الربط الديناميّ (DLL) لجهة المستضاف إستدعاءات تداخل البرمجة التطبيقيّة (API) للرسم في صفيفة حيث ترسلها في رسالة واحدة إلى الملقّم عندما تمتلىء الصفيفة أو عندما يدخل المستعمل دخلاً. راجع أيضاً تخبئة الصفة.

cache manager ـ برنامج إدارة المخبأ. مكوّن في نظام الدخل / الخرج يوفّر خدمات تخبئة الملفّات إلى أنظمة الملفّات ومغيّر الوجهة في النظام Windows NT. وهو يستعمل آليات الترتيب في صفحات لبرنامج إدارة الذاكرة الظاهريّة (VM) لإحضار الصفحات إلى الذاكرة من القرص وكتابة صفحات مخبّأة إلى القرص مجدّداً.

callback ـ الإستدعاء المسترد. رسالة طلب يرسلها ملقّم إلى مستضاف إستجابة لطلب من المستضاف، يرسل ملقّم إستدعاء مستردّ إلى مستضاف للحصول على معلومات إضافيّة. تتعلّق بطلب مستضاف. راجع إستدعاء الإجراء المحلي.

CDFS. نظام ملفّات CD-ROM.

child process ـ معالجة التابع. معالجة أُنشئت من قبل معالجة أخرى تسمى معالجة الأم. تتأصّل معالجة التابع بعض موارد معالجة الأم أو كلها. قارن مع معالجة الأم.

CISC. حاسوب مجموعة التعليمات المعقّدة.

client ـ المستضاف. معالجة ذات شعَب تستدعي الخدمات المتوفّرة من قبل إما معالجة ملقّم محلي أو عن بُعد. وفي النظام Windows NT، يتمّ الإتصال بين مستضاف وملقّم عبر

وسائل إستدعاء الإجراء المحلي (LPC) أو إستدعاء الإجراء عن بُعد (RPC). راجع أيضاً نموذج المستضاف / الملقّم وإستدعاء الإجراء المحلي وإستدعاء الإجراء عن بُعد.

client/server model ـــ نموذج المستضاف / الملقّم. نموذج لتصميم التطبيقات أو أنظمة التشغيل. يقسم النظام إلى معالجات (ملقّمات)، حيث توفّر كل منها مجموعة من الخدمات المتخصّصة للمعالجات الأخرى (المستضافات). تطلب معالجة المستضاف الخدمة بواسطة إرسال رسائل إلى معالجات الملقّم وترجع الملقّمات النتائج عبر رسالة أخرى. تناسب الأنظمة التي تعتمد على نموذج المستضاف / الملقّم لمحيطات الإحتساب الموزّع حيث تستطيع الملقّمات الإشتغال على حواسيب مختلفة.

code set ـــ مجموعة الشيفرة. الشيفرات الثنائيّة المستعملة لتمثيل أحرف لغة معيّنة.

committed memory ـــ الذاكرة المعتمدة. ذاكرة ظاهريّة حيث تمّ حجز فسحة في ملفّ الترتيب في صفحات. ويحدّد للمعالجة التي تعتمد الذاكرة حصة من ملفّ الترتيب في صفحات في ذلك الوقت. راجع أيضاً الذاكرة المحجوزة.

(CISC) complex instruction set computer ـــ حاسوب مجموعة التعليمات المعقّدة. معالج يستخدم تعليمات ماكنة قويّة متقنة. وبسبب تعقيدات التعليمات، يستغرق كل منها عدّة دورات ساعة للإتمام. قارن مع حاسوب مجموعة التعليمات المخفّضة.

concurrent application ـــ التطبيق المتزامن. تطبيق ينفّذ في موقعين أو أكثر. وفي النظام Windows NT، فإن التطبيق المتزامن هو الذي أنشأ أكثر من شعبة تنفيذ واحدة، إما ضمن معالجة واحدة أو في معالجات مستقلّة. راجع أيضاً المعالجة والشعبة.

configuration manager ـــ برنامج إدارة التشكيل. مجموعة من مكوّنات البراجيّات التي تبسّط تخزين المعلومات وتشكيل النظام وإستردادها. وهي تتضمّن مسجّل التشكيل ومحرّر المسجّل التخطيطي والبرامج الدائمة / براجيات مميّز العتاد. راجع أيضاً مسجّل التشكيل.

configuration registry ـــ مسجّل التشكيل. مستودع قاعدة بيانات للمعلومات المتعلّقة بتشكيل الحاسوب ــ مثلاً، عتاد الحاسوب والبراجيّات المركّبة على النظام وضوابط المحيط والمعلومات الأخرى المدخلة من قبل شخص أو أشخاص يستعملون النظام. راجع أيضاً الكائن المرجعي.

connecting an interrupt object ـــ توصيل كائن مقاطعة. ربط روتين خدمة مقاطعة (ISR) مع

مستوى طلب مقاطعة معيّن (IRQL). يستدعي مسيّق جهاز النظام لوصل كائن مقاطعة، الذي «يحفّز» مناولة المقاطعة للجهاز. راجع أيضاً فصل كائن مقاطعة وكائن المقاطعة وروتين خدمة المقاطعة.

console ــ الكونسول. إطار يعتمد على نص تحت إدارة النظام الفرعي Win 32. توجّه الأنظمة الفرعيّة للمحيط خرج تطبيقات في نمط الأحرف إلى الكونسول.

context ــ السياق. راجع سياق الشعبة.

context switching ــ التبديل السياقي. حفظ سياق شعبة تنفيذ، وتحميل سياق شعبة أخرى ونقل التحكُّم إلى شعبة جديدة. ينفّذ التبديل السياقي من قبَل موزّع النواة. راجع أيضاً الموزّع وسياق الشعبة.

control object ــ كائن التحكُّم. كائن نواة يوفّر طريقة نقّالة للتحكُّم بمهام النظام المتنوّعة. تتضمّن مجموعة كائنات التحكّم كائن إستدعاء الإجراء غير المتزامن (APC) وكائن إستدعاء الإجراء المؤجّل (DPC) وكائن معالجة النواة والكائنات المتعدّدة المستعملة من قبل نظام الدخل / الخرج. راجع أيضاً كائن النواة.

copy-on-write ــ النسخ عند الكتابة. حماية ذاكرة تعتمد على الصفحة (عكس التي تعتمد على الكائن) تتيح لمعالجتين مشاركة صفحة إلى أن تكتب إحداها إليها. في ذلك الوقت، يوفّر للمعالجة ذات الشعبة التي عدّلت الصفحة نسخة خاصة عن الصفحة في فسحة العنوان الظاهري العائدة لها.

critical section ــ القسم الحرج. مجموعة من الشيفرات التي تتمكّن من الوصول إلى مورد غير مشارك. ولتمكين الشيفرة الصحيحة، يمكن لشيفرة واحدة التنفيذ في القسم الحرج في كل مرة. راجع أيضاً المنع التبادلي.

(DPC) deferred procedure call ــ إستدعاء الإجراء المؤجّل. وظيفة تنفّذ لاتزامنياً، حيث تقاطع تنفيذ الشعبة الشغّالة حالياً. تنفّذ إستدعاءات DPC مهام النظام المؤجّلة إلى أن يهبط مستوى طلب المقاطعة (IRQL) لمعالج دون مستوى IRQL للتوزيع. راجع أيضاً كائن DPC وصفيفة DPC.

demand paging ــ الترتيب في صفحات عند الطلب. سياسة إحضار للترتيب في صفحات التي تؤجّل تحميل الصفحات في الذاكرة الفعليّة إلى أن يحصل خطأ صفحة. راجع أيضاً سياسة الإحضار.

desired access rights ـ حقوق الوصول المطلوبة. مجموعة حقوق الوصول التي تطلبها شعبة عند فتح مقبض إلى كائن. راجع أيضاً حقوق الوصول الممنوحة.

device object ـ كائن الجهاز. كائن نظام يمثّل جهازاً فعلياً أو منطقياً أو ظاهرياً ويصف خصائصه. يتعلق كائن جهاز مع كائن مسيّق. راجع أيضاً كائن المسيّق.

disconnecting an interrupt object ـ فصل كائن مقاطعة. فكّ إرتباط روتين خدمة مقاطعة (ISR) عن مستوى طلب مقاطعة (IRQL) معيّن. يستدعي مسيّق جهاز النظام لفصل كائن مقاطعة، الذي «يوقف» مناولة المقاطعة للجهاز. راجع أيضاً توصيل كائن مقاطعة وكائن المقاطعة وروتين خدمة المقاطعة.

discretionary access control ـ التحكّم بالوصول الإستنسابي. تطبّق حماية مالك كائن على الكائن عن طريق تعيين حقوق وصول متنوّعة إلى المستعملين أو إلى مجموعة من المستعملين. ويمكن تحديد الحماية الإستنسابيّة إلى التحكّم بالوصول الإجباري المطبّق على الكائن. راجع أيضاً التحكُّم بالوصول الإجباري.

disk mirroring ـ إنشاء نسخ مرآويّة عن القرص. إجراء إستنساخ تقسيم قرص على قرصين أو أكثر، ويفضّل على أقراص مثبّتة إلى وحدات التحكُّم بالقرص مستقلّة لكي تبقى البيانات قابلة للوصول عند إخفاق قرص أو وحدة تحكُّم بالقرص. راجع أيضاً السماح بالأعطال.

disk striping ـ تجريد القرص. طريقة دمج مجموعة من تقسيمات القرص من نفس الحجم الموجودة على أقراص منفصلة إلى واحد يشكّل «شريطاً» ظاهرياً عبر الأقراص. تمكّن هذه الطريقة عمليات الدخل / الخرج المتعدّدة في نفس الحجم من المتابعة بتزامن. راجع تجريد القرص مع التماثل.

disk striping with parity ـ تجريد القرص مع التماثل. طريقة المحافظة على معلومات التماثل عبر شريطة قرص بحيث في حال تعطّل تقسيم قرص واحد، يمكن إعادة إنشاء البيانات على ذلك القرص. راجع أيضاً تجريد القرص والسماح بالأعطال.

dispatcher ـ الموزّع. منظومة نواة تتعقّب الشعَب الجاهزة للتنفيذ وتحدّد ترتيب تشغيلها وتحفّز التبديل السياقي من شعبة واحدة إلى أخرى. راجع أيضاً التبديل السياقي.

dispatcher database ـ قاعدة بيانات الموزّع. مجموعة من بنيات البيانات الشاملة التي تستعملها النواة لتعقّب الشعَب الجاهزة للتنفيذ والمعالجات التي تنفّذ الشعَب. تتضمّن قاعدة البيانات صفيفة جهوزيّة الموزّع. راجع أيضاً الموزّع وصفيفة جهوزيّة الموزّع.

dispatcher object ـ **كائن الموزّع**. كائن نواة يدعم المزامنة. يستخدم موزّع النواة السنيّات المزامنة المؤشّرة وغير المؤشّرة. راجع أيضاً كائن النواة والحالة المؤشّرة والحالة غير المؤشّرة.

dispatcher ready queue ـ **صفيفة جهوزيّة الموزّع**. بنية البيانات في قاعدة بيانات الموزّع التي تتعقّب الشعَب الجاهزة للتنفيذ. وهي سلسلة من الصفيفات، صفيفة واحدة لكل أولويّة مجدولة. راجع أيضاً الموزّع وقاعدة بيانات الموزّع.

domain controller ـ **وحدة التحكّم بالميدان**. ملقّم في ميدان شبكة يقبل تسجيلات المستعمل ويحفز التحقّق من أصالتها. راجع أيضاً التحقّق من الأصالة.

DPC. إستدعاء الإجراء المؤجّل.

DPC object ـ **كائن إستدعاء الإجراء المؤجّل**. كائن نواة مستعمل لينفّذ لاتزامنياً وظيفة نظام. وهي كائن تحكّم يحتوي عنوان إستدعاء إجراء مؤجّل (DPC) للتنفيذ. تضع النواة كائنات DPC في صفيفة DPC شاملة لتنتظر التنفيذ. راجع أيضاً إستدعاء الإجراء المؤجّل وصفيفة DPC.

DPC queue ـ **صفيفة إستدعاء الإجراء المؤجّل**. بنية بيانات بإدارة النواة تحتوي إستدعاءات إجراء مؤجّل (DPCs) تنتظر التنفيذ. يؤدّي وجود كائن DPC في صفيفة DPC إلى إصدار النواة مقاطعة براجيات عند مستوى طلب المقاطعة (IRQL) للتوزيع / DPC. ويقوم المعالج الذي يستلم المقاطعة بنقل التحكّم إلى النواة التي تنفّذ كل إستدعاءات DPC في الصفيفة. راجع أيضاً إستدعاء الإجراء المؤجّل وكائن DPC.

driver object ـ **كائن المسيّق**. كائن نظام يمثّل مسيّقاً إفرادياً على النظام ويبلغ إلى برنامج إدارة الدخل / الخرج عنوان نقاط إدخال المسيّق. ويمكن أن يتعلّق كائن المسيّق مع كائنات الجهاز المتعدّدة. (يمثّل كل منها جهاز يشغله المسيّق). راجع أيضاً كائن الجهاز.

(DLL) dynamic-link library ـ **مكتبة الربط الديناميّ**. روتين تداخل برمجة تطبيقيّة (API) تستعمله التطبيقات في نمط المستعمل للوصول إلى إستدعاءات الإجراء العادي. ولا تشمل شيفرة روتين API في الرسم المنفّذ للمستعمل. لكن وعِوَضاً عن ذلك، يعدّل تلقائياً نظام التشغيل الرسم المنفّذ ليشير إلى إجراءات DLL عند وقت التشغيل.

environment subsystem ـ **النظام الفرعي للمحيط**. نظام فرعي محمي (ملقّم) يوفّر روتين تداخل برمجة تطبيقيّة (API) ومحيطاً ـ مثل Win 32 و MS-DOS و POSIX أو OS/2 ـ على النظام Windows NT. يلتقط هذا النظام الفرعي إستدعاءات API ويستخدمها عن طريق

إستدعاء خدمات Windows NT المحليّة. راجع أيضاً النظام الفرعي المتكامل والنظام الفرعي المحمي.

exception ـ **الإستثناء**. حالة خطأ متزامن ناتجة عن تنفيذ تعليمة ماكنة معيّنة. يمكن أن تكون الإستثناءات أخطاء مكتشفة من قبل العتاد كالقسمة على صفر أو أخطاء مكتشفة من قبل البراجيّات مثل مخالفة صفحة وقاية. راجع أيضاً مناول الإستثناءات والمناولة الإستثنائيّة البنيويّة ومناول المصيدة.

exception dispatcher ـ **موزّع الإستثناءات**. منظومة نواة توزّع الإستثناءات وتنقل التحكُّم إلى مناولات الإستثناءات المزوّدة من قبَل المستدعي أو، في حال عدم أي منها، تنفيذ مناولات الإستثناءات المفترضة للنظام. راجع أيضاً الإستثناءات ومناول الإستثناءات.

exception handler ـ **مناول الإستثناءات**. الشيفرة التي تستجيب للإستثناءات والنوعان هما مناولاً الإستثناءات الإطاريّة (بما في ذلك مناولات الإنتهاء) ومناولات الإستثناءات المفترضة للنظام. راجع أيضاً الإستثناء والمناولة الإستثنائيّة البنيويّة ومناول الإنهاء.

executive ـ البرنامج التنفيذي. راجع البرنامج التنفيذي NT.

executive object ـ الكائن التنفيذي. كائن NT مرئي لنمط المستعمل بواسطة مكوّن البرنامج التنفيذي NT. يصدّر البرنامج التنفيذي NT خدمات الكائن المستعملة لمناولة الكائنات التنفيذيّة.

FAT. جدول تحديد مواقع الملفّات.

FAT file system ـ نظام الملفّات FAT. نظام الملفّات المستعمل على أنظمة MS-DOS.

fault tolerance ـ السماح بالأعطال. قدرة حاسوب ونظام تشغيل في الإستجابة على أحداث مدمّرة مثل إنقطاع الطاقة الكهربائيّة أو تعطّل العتاد. يطبّق السماح بالأعطال عادة القدرة على متابعة تشغيل النظام دون فقد البيانات أو توقيف النظام وإعادة تشغيله حيث تستعاد المعالجة التي كانت قيد التنفيذ عند حصول العطل.

fetch policy ـ سياسة الإحضار. اللوغاريتم التي تستعمله ذاكرة ظاهريّة لتحديد ناقل الصفحات الذي يجب أن يحضّر صفحة من القرص إلى الذاكرة. يستعمل النظام Windows NT لوغاريتم ترتيب الطلبات في صفحات معدّل. راجع أيضاً ترتيب الطلبات في صفحات.

file handle ـ مقبض الملفّ. مقبض لكائن ملفّ. راجع أيضاً كائن الملفّات.

file-mapping object ــ كائن تخطيط الملفّات. إصدار النظام الفرعي Win 32 لكائن قسم NT مدعوم من قبل ملفّ مخطّط.

file object ــ كائن الملفّات. كائن تنفيذي يمثّل ملفاً مفتوحاً ودليلاً ومجلّداً أو جهازاً. راجع أيضاً الكائن التنفيذي.

frame-based exception handler ــ مناول الإستثناءات الإطاري. مناول إستثناء متعلّق بإجراء معيّن أو قسم من إجراء. تحفّز النواة مناول إستثناء إطاري عند حصول إستثناء ضمن كتلة الشيفرة. ويمكن أن يعالج مناول الإستثناء الإطاري الإستثناء أو يحوّله إلى طبقة شيفرة أعلى أو يتجاهل الإستثناء ويتابع تنفيذ البرنامج. راجع أيضاً الإستثناء والمناولة الإستثنائيّة البنيويّة ومناول الإنهاء.

granted access rights ــ حقوق الوصول الممنوحة. مجموعة حقوق الوصول التي يمنحها نظام الأمان إلى شعبة تفتح مقبضاً إلى كائن. وحقوق الوصول الممنوحة هي مجموعة فرعيّة غير صحيحة للوصول المطلوب من قبل الطالب. يخزّن برنامج إدارة الكائنات حقوق الوصول الممنوحة في مقبض الكائن الذي يرجعه. راجع أيضاً حقوق الوصول المطلوبة.

HAL. طبقة تجريد العتاد.

handle ــ مقبض. راجع مقبض الكائن.

hardware abstraction layer (HAL) ــ طبقة تجريد العتاد. مكتب ربط دينامي (DLL) تحمي البرنامج التنفيذي NT من التغييرات في منصّات عتاد البائعين المختلفة وذلك لزيادة نقليّة نظام التشغيل إلى الحدّ الأقصى. تستخدم الطبقة HAL الوظائف التي تجرّد تداخلات الدخل / الخرج ووحدة التحكّم بالمقاطعة ومخابىء العتاد وآليات الإتصال للمعالج المتعدّد وما شابه.

high performance file system (HPFS) ــ نظام الملفّات العالي الأداء. نظام ملفّات مصمّم للنظام OS/2 الإصدار 1.2، الذي أنشىء لعَنونة حدود نظام ملفّات جدول تحديد مواقع الملفّات (FAT) المستعمل من قبل النظام MS-DOS. وقد أضاف مزايا مثل أسماء ملفّات أطول والقدرة على ربط الصفات بالملفّ والبحث الأسرع عن الملفّات والإستمثالات الأخرى.

HPFS. نظام الملفّات العالي الأداء.

Idle thread ــ الشعبة المتوقّفة. شعبة نظام تنفّذ عندما لا تكون أية شعبة أخرى جاهزة

للتنفيذ. تنفّذ الشعبة المتوقّفة إستدعاءات الإجراء المؤجّل (DPC) وتحفّز التبديل السياقي عندما تصبح شعبة أخرى جاهزة للتنفيذ. ويوجد شعبة واحدة متوقّفة لكل معالج في نظام معالج متعدّد.

IDT. جدول توزيع المقاطعة.

IFS. نظام الملفّات القابل للتركيب.

impersonation ـ التقليد. قدرة شعبة في معالجة واحدة على تقليد الأمان لشعبة في معالجة أخرى وتنفيذ العمليّات نيابة عن الشعبة. وهي تستعمل من قبل الأنظمة الفرعيّة للمحيط وخدمات الشبكة عند الوصول إلى موارد عن بعد لتطبيقات المستضاف.

(IFS) installable file system ـ نظام الملفّات القابل للتركيب. نظام ملفّات يمكن تحميله في نظام التشغيل ديناميا. يستطيع النظام Windows NT دعم أنظمة ملفّات قابلة للتركيب متعدّدة في نفس الوقت، بما في ذلك نظام ملفّات جدول تحديد موقع الملفّات (FAT) ونظام ملفّات NT (NTFS) ونظام الملفّات العالي الأداء (HPFS) ونظام ملفّات (CDFS) CD-ROM. يحدّد نظام التشغيل تلقائياً نسق وسيط التخزين ويقرأ ويكتب الملفّات في النسق الصحيح.

instruction execution unit ـ وحدة تنفيذ التعليمات. كتلة تعتمد على معالج للشيفرات في ماكنة DOS ظاهريّة (VDM). وهي تعمل كمناول مصيدة على معالجات Intel وكمؤهّل تعليمات Intel على معالجات MIPS. راجع أيضاً ماكنة DOS الظاهريّة.

integral subsystem ـ النظام الفرعي المتكامل. نظام فرعي محمي (ملقّم) ينفّذ مهمة نظام تشغيل أساسيّة. تتضمّن هذه المجموعة ملقّمات الشبكة والنظام الفرعي للأمان. راجع أيضاً النظام الفرعي للمحيط والنظام الفرعي المحمي.

interrupt ـ المقاطعة. حالة نظام تشغيل غير متزامنة تقاطع التنفيذ الإعتيادي وتنقل التحكّم إلى مناول مقاطعة. تحفّز عادة المقاطعات بواسطة أجهزة الدخل / الخرج التي تتطلّب خدمة من المعالج. راجع أيضاً الإستثناء ومناول المصيدة.

interrupt dispatcher ـ موزّع المقاطعة. منظومة فرعيّة لمناول مصيدة النواة. وهي تحدّد مصدر المقاطعة وتنقل التحكّم إلى روتين يتناول المقاطعة. راجع أيضاً المقاطعة.

(IDT) interrupt dispatch table ـ جدول توزيع المقاطعة. بنية بيانات لكل معالج تستعملها

النواة لتحديد موقع روتين مناولة مقاطعة عند حصول مقاطعة. راجع أيضاً موزّع المقاطعة.

interrupt object ـ كائن المقاطعة. كائن نواة يتيح لمسيّق جهاز ربط («وصل») روتين خدمة مقاطعة (ISR) مع مستوى طلب مقاطعة (IRQL). وهو كائن تحكّم يحتوي عنوان الروتين ISR والمستوى IRQL حيث يقاطع الجهاز والإدخال في جدول توزيع المقاطعة (IDT) مع الروتين ISR الذي يوصل إليه. راجع أيضاً توصيل كائن مقاطعة، وجدول توزيع المقاطعة، ومستوى طلب المقاطعة، وروتين خدمة المقاطعة.

interrupt request level (IRQL) ـ مستوى طلب المقاطعة. ترتيب المقاطعات وفقاً للأولويّة. يحتوي المعالج على ضبط مستوى طلب مقاطعة (IRQL) تستطيع الشعبة رفعه أو تخفيضه. تمنع المقاطعات التي تحصل عند أو دون ضبط المستوى IRQL للمعالج أو تحجب بينما لا تحجب المقاطعات التي تحصل فوق ضبط المستوى IRQL للمعالج. راجع أيضاً حجب المقاطعات.

interrupt service routine (ISR) روتين خدمة المقاطعة. روتين مسيّق جهاز الذي يستدعيه مناول مقاطعة النواة عندما يصدر جهاز مقاطعة. يدقّق روتين الجهاز عن توليد المقاطعات ويحفظ معلومات حالة الجهاز ثم يضع في صفيفة إستدعاء الإجراء المؤجّل (DPC) لمسيّق الجهاز لإتمام خدمة المقاطعة. راجع أيضاً إستدعاء الإجراء المؤجّل.

invalid page ـ صفحة غير صالحة. صفحة ظاهريّة تؤدّي إلى خطأ صفحة إذا تمّ الرجوع إلى عنوان منها. وتكون الصفحة إمّا محمّلة من قرص وتجعل صالحة أو تُستعاد من قائمة صفحات إحتياط أو معدّلة وتجعل صالحة. وإلا، فالمرجع يكون مخالفة في الوصول. راجع أيضاً الصفحة الصالحة.

I/O completion ـ إتمام الدخل / الخرج. الخطوة الأخيرة في معالجة نظام الدخل / الخرج لطلب دخل / خرج. تشمل العمليّات النموذجيّة حذف بنيات البيانات الداخليّة المتعلّقة بالطلب وإرجاع البيانات إلى المستوى وتسجيل الحالة النهائيّة للعمليّة في كتلة حالة الدخل / الخرج وضبط كائن ملفّ و / أو حدث إلى الحالة المؤشّرة وربما وضع في صفيفة إستدعاء إجراء غير متزامن (APC). راجع أيضاً إستدعاء الإجراء غير المتزامن.

I/O manager ـ برنامج إدارة الدخل / الخرج. مكوّن البرنامج التنفيذي Windows NT الذي يوحّد الأجزاء المختلفة من نظام الدخل / الخرج. وهو يعرّف إطار عمل منظّم حيث تقبل طلبات الدخل / الخرج وتسلّم إلى أنظمة الملفّات ومسيّقات الجهاز. وهو يوفّر أيضاً الشيفرة العامة لأكثر من جهاز واحد.

I/O request packet (IRP) ــ حزمة طلب الدخل / الخرج. بنية بيانات مستعملة لتمثّل طلب دخل / خرج وللتحكّم بمعالجته. ينشىء برنامج إدارة الدخل / الخرج الحزمة IRP ثم يمرّرها إلى مسيّق واحد أو أكثر في تتابع. وعند إنتهاء المسيّقات من تنفيذ العمليّة، يتمّم برنامج إدارة الدخل / الخرج طلب الدخل / الخرج ويحذف الحزمة IRP. راجع أيضاً إتمام الدخل / الخرج.

IOSB. كتلة حالة الدخل / الخرج.

I/O status block (IOSB) ــ كتلة حالة الدخل / الخرج. بنية بيانات يزوّدها مستدعي كبارامتر إلى خدمة دخل / خرج. يسجّل برنامج إدارة الدخل / الخرج الحالة النهائيّة للعمليّة في كتلة حالة الدخل / الخرج عند إتمام المعالجة.

IRP. حزمة طلب الدخل / الخرج.

IRP stack location ــ موقع تكديس حزمة طلب الدخل / الخرج. منطقة بيانات حزمة طلب دخل / خرج تحتوي معلومات يحتاجها مسيّق معيّن لتنفيذ القسم العائد له من طلب الدخل / الخرج. يحتوي كل مسيّق يعمل على الطلب موقع تكديس مستقلّ في الحزمة IRP. راجع أيضاً حزمة طلب الدخل / الخرج.

IRQL. مستوى طلب المقاطعة.

ISR. روتين خدمة المقاطعة.

kernel. راجع النواة NT.

kernel mode ــ نمط النواة. نمط المعالج بأفضليّة حيث تستطيع شيفرة النظام Windows NT الإشتغال. تحتوي شعبة تشتغل في نمط النواة على الوصول إلى ذاكرة النظام وإلى العتاد. قارن مع نمط المستعمل.

kernel object ــ كائن النواة. حالة آنية لوقت تشغيل نوع بيانات مجرّدة معرّفة من قبل النواة. تعرّف النواة ألسنيّة خاصة لتصرّف كائنات النواة وتستخدم روتينات النواة التي يستطيع البرنامج التنفيذي NT إستدعاءها لمناولة كائنات النواة. تصنّف كائنات النواة في فئتين: كائنات التحكّم وكائنات الموزّع. يستعمل نوعا كائنات النواة كمرجع لكائنات البرنامج التنفيذي Windows NT. راجع أيضاً كائن التحكّم وكائن الموزّع والكائن التنفيذي.

kernel process object ــ كائن معالجة النواة. عرض النواة لمعالجة. وهو كائن تحكّم يحتوي

المعلومات الضروريّة لتحميل فسحة عنوان النواة وتعقّب موارد المعالجة والصفات المفترضة. راجع أيضاً كائن التحكّم.

kernel thread object ـ كائن شعبة النواة. عرض النواة لشعبة. وهو كائن موزّع يحتوي المعلومات الأوليّة الضروريّة لتوزيع الشعبة للتنفيذ. راجع أيضاً كائن الموزّع.

key object ـ الكائن المرجعي. كائن تنفيذي يمثّل معلومات تشكيل النظام المخزّنة في مسجّل التشكيل. راجع أيضاً مسجّل التشكيل والكائن التنفيذي.

lazy evaluation algorithms ـ خوارزميّة التقييم المقتصدة. فئة عامة للخوارزميّات التي تتجنّب تنفيذ عمليّة مكلفة إلى أن تطلب. وإذا لم تطلب أبداً العمليّة، لا يهدر أي وقت معالجة. يستعمل برنامج إدارة الذاكرة الظاهريّة (VM) في Windows NT خوارزميّات التقييم المقتصدة لتحسين أداء الذاكرة. إن الترتيب في صفحات عند الطلب وحماية الصفحة بالنسخ عند الكتابة وحفظ واعتماد الذاكرة بشكل مستقلّ هي أمثلة عن ذلك. راجع أيضاً الذاكرة المعتمدة والنسخ عند الكتابة والترتيب في صفحات عند الطلب والذاكرة المحجوزة.

locale ـ السوق المحلي. المحيط الوطني و/ أو المحلي حيث يشتغل نظام أو برنامج. يحدّد السوق المحلي اللغة المستعملة للرسائل والقوائم وترتيب فرز النضائد وتصميم لوحة المفاتيح والمصطلحات وتنسيق التاريخ والوقت.

(LPC) local procedure call facility ـ وسيلة إستدعاء الإجراء المحلي. وسيلة تمرير رسائل مثلى تتيح لمعالجة واحدة الإتصال مع معالجة أخرى على نفس الماكنة. تستعمل الأنظمة الفرعيّة المحميّة وسيلة LPC للإتصال مع بعضها البعض ومع معالجات المستضاف. الوسيلة LPC هي وسيلة مغيّرة لنموذج تمرير الرسائل لإستدعاء الإجراء عن بُعد (RPC) ومستمثلة للإستعمال المحلي. قارن مع إستدعاء الإجراء عن بُعد.

local replacement policy ـ سياسة الإستبدال المحليّة. خوارزميّة إستبدال صفحة تحدّد موقع عدد ثابت من أُطُر الصفحات إلى كل معالجة. وعندما تتجاوز معالجة حصتها، يبدأ برنامج إدارة الذاكرة الظاهريّة (VM) بنقل الصفحات في مجموعة عمل المعالجة إلى قرص لتخلية فسحة لأخطاء صفحة إضافيّة تولّدها المعالجة. راجع أيضاً سياسة الإستبدال ومجموعة العمل.

logon process ـ معالجة التسجيل. معالجة Windows NT ذات شعَب تكتشف محاولة مستعمل

التسجيل على نظام التشغيل. وهي تتحقّق من معلومات تسجيل المستعمل مع نظام الأمان قبل منح المستعمل الوصول إلى النظام.

LPC. إستدعاء الإجراء المحلي.

mandatory access control ـ **التحكُّم بالوصول الإجباري**. حماية معيّنة إلى كائن من قبل مدير النظام. توسم عادة بنود التحكّم بالوصول الإجباريّة الكائنات بمستوى، مثل «سرّي» أو «سرّي جداً». ويجب على المستعمل الذي يرغب بالوصول إلى الكائنات أن يتجاوز المستوى المناسب. يحل التحكّم بالوصول الإجباري محل بنود التحكّم بالوصول الإستنسابي التي يطبّقها المالك على كائن. راجع أيضاً التحكّم بالوصول الإستنسابي.

map ـ **الخريطة**. لترجمة عنوان ظاهري إلى عنوان فعلي.

mapped file ـ **الملفّ المخطّط**. ملفّ محمّل في كائن قسم في الذاكرة. وبواسطة تخطيط معاينات القسم في فسحة عنوانها، تستطيع معالجة الوصول إلى الملفّ كصفيفة كبيرة مخزّنة في الذاكرة الظاهريّة. ويعينّ برنامج إدارة الذاكرة الظاهريّة (VM) تلقائياً الصفحات من وإلى الملفّ، حيث يحمّل الصفحات من القرص عند إستعمالها ويكتب الصفحات إلى القرص عند تعديلها. راجع أيضاً الخريطة وترتيب صفحات الملفّ.

mapped file I/O ـ **دخل / خرج الملفّ المخطّط**. دخل / خرج ملفّ منفّذ بواسطة القراءة والكتابة إلى الذاكرة الظاهريّة المدعومة من قبَل ملفّ. راجع أيضاً الملفّ المخطّط.

marshal ـ **المنظّم**. لترتيب وتوضيب بارامترات الإجراء في نسق معينّ للإرسال عبر الشبكة. راجع أيضاً إستدعاء الإجراء من بُعد.

masking interrupts ـ **حجب المقاطعات**. رفع مستوى طلب مقاطعة (IRQL) لمعالج لمنع المقاطعات عند مستوى IRQL الجديد ودونه.

master/slave system ـ **النظام الرئيسي / الثانوي**. راجع المعالجة المتعدّدة غير المتناظرة.

messenger service ـ **خدمة المرسل**. خدمة شبكة تستلم الرسائل من الأنظمة الأخرى وتعرضها. راجع أيضاً الخدمة.

method ـ **الطريقة**. وظيفة تتعلّق بنوع كائن يستدعيه برنامج إدارة الكائنات تلقائياً عند نقاط معرّفة بشكل جيّد خلال مدة خدمة كائن. راجع أيضاً نوع الكائن.

MIDL ـ لغة تعريف التداخل من Microsoft.

MIDL compiler ـ مصرّف لغة تعريف من **Microsoft**. مصرّف يستلم ملفّات مكتوبة بلغة تعريف التداخل من Microsoft ويصدّر روتينات فرعيّة للإستعمال في تطبيقات إستدعاء الإجراء عن بُعد (RPC). راجع أيضاً إستدعاء الإجراء عن بُعد والإجراء الفرعي.

modified page writer ـ كاتب الصفحة المعدّلة. شعبة برنامج في إدارة الذاكرة الظاهريّة (VM) التي تكتب لاتزامنياً صفحات ظاهريّة معدّلة إلى قرص، حيث تزيد عدد أُطُر الصفحة المتوفّرة.

MPR. مسلّك الموفّر المتعدّد.

(MPR) multiple provider router ـ مسلّك الموفّر المتعدّد. مكتبة ربط دينامي (DLL) تحدّد الشبكة (وبالتالي نظام الملفّات) للوصول عندما يستعمل تطبيق تداخل البرمجة التطبيقيّة (API) في Win 32 WNet لتصفّح أنظمة الملفّات عن بُعد.

(MUP) multiple UNC provider ـ موفّر مصطلح التسمية المتناسق المتعدّد. مسيّق يحدّد الشبكة (وبالتالي نظام الملفّات) للوصول عندما يستعمل تطبيق تداخل البرمجة التطبيقيّة (API) في الدخل / الخرج Win 32 لفتح الملفّات عن بُعد.

multiprocessing ـ المعالجة المتعدّدة. تنفيذ نظام التشغيل في نفس الوقت لشعبتين أو أكثر على معالجات مختلفة. وتستطيع فقط أنظمة التشغيل المتعدّدة المعالجة إستخدام المعالجات الإضافيّة في حاسوب متعدّد المعالج. وكقاعدة عامة، تنفّذ أيضاً أنظمة التشغيل المتعدّدة المعالجة تنفيذ المهام المتعدّدة. راجع أيضاً المهام المتعدّدة.

multiprogramming ـ البرمجة المتعدّدة. راجع المهام المتعدّدة.

multitasking ـ المهام المتعدّدة. تنفيذ معالج لأكثر من شعبة واحدة بالتبديل السياقي من واحدة إلى أخرى، حيث يوهم أن كل الشعَب تنفّذ في نفس الوقت. راجع أيضاً المهام المتعدّدة الشفعيّة.

multithreading ـ الشعَب المتعدّدة. قدرة تطبيق على التنفيذ في موقفين أو أكثر بإستعمال شعَب متعدّدة. ويستعمل التعبير أحياناً بالتوافق مع المهام المتعدّدة للإشارة إلى نظام تشغيل يدعم الشعَب.

MUP. موفّر مصطلح التسمية المتناسق المتعدّد (UNC).

mutual exclusion ـ المنع المتبادل. الإتاحة لشعبة واحدة فقط في كل مرة للوصول إلى مورد.

المنع المتبادل ضروري عندما لا يعير مورد نفسه إلى الوصول المشارك أو عند إمكانيّة قيام المشاركة بإنشاء نتائج غير متوقّعة. راجع أيضاً القسم الحرج.

named pipe ـ **الأنبوب المسمّى**. آلية إتصال بين المعالجات تتيح لمعالجة واحدة إرسال البيانات إلى موقع آخر أو إلى معالجة عن بُعد.

name retention ـ **حجز الإسم**. الإجراء الذي يعتمده برنامج إدارة الكائنات لإبقاء إسم كائن في فسحة إسمه. وعند إغلاق آخر مقبض إلى الكائن، يحذف برنامج إدارة الكائنات إسم الكائن من فسحة عنوانه حيث يمنع عمليات الفتح اللاحقة على ذلك الكائن. راجع أيضاً إحتجاز الكائن.

(NLS) national language support ـ **دعم اللّغة الوطنيّة**. تداخل برمجة تطبيقيّة (API) توفّر للتطبيقات الوصول إلى معلومات خاصة بالسوق المحلي. راجع أيضاً السوق المحلي.

native services ـ **الخدمات المحليّة**. خدمات النظام التي يوفّرها البرنامج التنفيذي NT لنمط المستعمل لتستعمل من قبل الأنظمة الفرعيّة للمحيط ومكتبات الربط الديناميّ (DLL) وتطبيقات النظام الأخرى.

NDIS. مواصفات التداخل لمسيّق الشبكة.

NetBEUI transport ـ **النقل في NetBEUI**. تداخل المستعمل الموسّع (نظام الدخل / الخرج الأساسي للشبكة) NetBIOS. بروتوكول نقل الشبكة في المنطقة المحليّة الأوليّة في النظام Windows NT. راجع أيضاً تداخل NetBIOS.

NetBIOS interface ـ **تداخل نظام الدخل / الخرج الأساسي للشبكة**. تداخل برمجة يتيح إرسال طلبات الدخل / الخرج إلى حاسوب عن بُعد وإستلامها منه. وهو يخفي عتاد تشبيك الحواسيب عن التطبيقات.

network domain ـ **ميدان الشبكة**. مجموعة من محطات العمل والملقّمات التي تشارك قاعدة بيانات برنامج إدارة حسابات الأمان (SAM) ويمكن إدارتها كمجموعة. يستطيع مستعمل ذات حساب في ميدان شبكة معيّن التسجيل والوصول إلى حسابه من أي نظام في الميدان. راجع أيضاً قاعدة بيانات SAM.

(NDIS) network driver interface specification ـ **مواصفات التداخل لمسيّق الشبكة**. تداخل Windows NT لمسيّقات بطاقة الشبكة. وهو يوفّر إستقلاليّة النقل لبائعي بطاقات الشبكة لأن كل مسيّقات النقل تستطيع إستدعاء تداخل NDIS للوصول إلى بطاقات الشبكة.

وتكون مسيّقات الشبكة المكتوبة إلى تداخل NDIS (مسيّقات NDIS) نقّالة إلى محيط مسيّق الجهاز الظاهري في MS-DOS.

network redirector ـ مغيّر وجهة الشبكة. براجيّات تشبيك الحواسيب التي تقبل طلبات الدخل / الخرج للملفّات عن بُعد، وأنابيب مسمّاة أوشقوق بريديّة وترسلها («تغيّر وجهتها») إلى ملقّم شبكة على ماكنة أخرى. تستخدم مغيّرات الوجهة كمسيّقات لنظام الملفّات في Windows NT. راجع أيضاً ملقّم الشبكة.

network server ـ ملقّم الشبكة. براجيّات شبكة تستجيب لطلبات دخل / خرج أو إحتساب من ماكنة مستضاف. يمكن إستخدام ملقّمات شبكة Windows NT إما كمعالجات ملقّم أو كمسيّقات. راجع أيضاً النظام الفرعي المحمي.

NLS. دعم اللغة الوطنيّة.

nonpaged pool ـ مجموعة غير مرتّبة في صفحات. قسم من ذاكرة النظام الذي لا يمكن ترتيبه في صفحات إلى قرص. قارن مع المجموعة المرتّبة في صفحات.

nonprivileged processor mode ـ نمط المعالج بدون أفضليّة. راجع نمط المستعمل.

nonsignaled state ـ الحالة غير المؤشّرة. صفة كل كائن بنوع كائن يدعم المزامنة. تواصل شعبة تنتظر كائناً، أي موجودة في حالة غير مؤشّرة، الإنتظار إلى أن تضبط النواة الكائن إلى الحالة المؤشّرة. راجع أيضاً الحالة المؤشّرة والمزامنة.

NT ـ التكنولوجيا الحديثة.

NT executive ـ البرنامج التنفيذي NT. قسم من نظام التشغيلWindows NT الذي يشتغل في نمط النواة. وهو يوفّر بنية المعالجة والإتصال بين المعالجات وإدارة الذاكرة وإدارة الكائن وجدولة الشعَب ومعالجة المقاطعات وقدرات الدخل / الخرج وتشبيك الحواسيب وأمان الكائن. تتوفّر تداخلات البرمجة التطبيقيّة (API) والمزايا الأخرى في الأنظمة الفرعيّة المحميّة في نمط المستعمل. راجع أيضاً النظام الفرعي المحمي.

(NTFS) NT file system ـ نظام الملفّات NT. نظام ملفّات متقدّم مصمّم للإستعمال خصوصاً مع نظام التشغيل Windows NT. وهو يدعم إستعادة نظام الملفّات ووسيط تخزين كبير جداً ومزايا متنوّعة للنظام الفرعي POSIX. وهو يدعم أيضاً التطبيقات الكائنيّة عن طريق معالجة كل الملفّات ككائنات مع الصفات المعرّفة من قبل المستعمل والمعرّفة من قبل النظام.

NTFS ـ نظام الملفّات NT.

NT kernel ـ **النواة NT. مكوّن البرنامج التنفيذي NT** الذي يدير المعالج. وهو ينفّذ جدولة وتوزيع الشعَب ومناولة المقاطعة والإستثناء ومزامنة المعالجات المتعدّدة، ويوفّر الكائنات الأوليّة التي يستعملها البرنامج التنفيذي NT لإنشاء كائنات نمط المستعمل.

object ـ **الكائن**. حالة آنية واحدة لوقت تشغيل لنوع كائن معرّف من قبل NT. وهو يحتوي بيانات يمكن مناولتها فقط بإستعمال مجموعة من الخدمات المتوفّرة للكائنات من نوعه. راجع أيضاً نوع الكائن.

object attribute ـ **صفة الكائن**. حقل بيانات في كائن يعرّف أو يسجّل حالة الكائن والذي يمكن مناولته بإستدعاء خدمة كائن.

object class ـ **فئة الكائن**. راجع نوع الكائن.

object directory object ـ **كائن دليل الكائن**. كائن يخزّن أسماء الكائنات الأخرى، كما يخزّن دليل الملفّات. وهو يوفّر الوسائل لدعم بنية تسمية تسلسليّة لكائنات Windows NT.

object domain ـ **ميدان الكائن**. مجموعة ذاتيّة الإحتواء من الكائنات التي يمكن الوصول إليها بواسطة التسلسل الهرمي لإسم كائن برنامج إدارة الكائنات NT لكنه يُدار بواسطة برنامج إدارة كائن ثانوي (مثل نظام الدخل / الخرج NT).

object handle ـ **مقبض الكائن**. فهرس في جدول كائنات خاص لمعالجة. وهو يستعمل ليشير إلى كائن مفتوح ويشمل مجموعة من حقوق الوصول الممنوحة إلى المعالجة التي تملك المقبض. وهو يحتوي أيضاً تسمية تأصّل تحدّد تأصّل المقبض من قبل معالجات التابع. تستعمل البرامج المقابض للإشارة إلى الكائنات عند إستدعاء خدمات الكائن. راجع أيضاً جدول الكائنات.

object manager ـ **برنامج إدارة الكائنات**. مكوّن البرنامج التنفيذي NT الذي ينشىء موارد نظام التشغيل ويحذفها ويسمّيها، المخزّنة ككائنات.

object model ـ **نموذج الكائن**. نموذج لتصميم البرامج حول البيانات التي تتناولها. تخفى بنيات البيانات داخل الكائنات ويجب أن تستعمل البرامج الخدمات المعرّفة بشكل خاص لمناولة بيانات الكائن. إن الهدف الرئيسي لنموذج الكائن هو زيادة إمكانيّة إعادة إستعمال الشيفرة إلى الحدّ الأقصى. راجع أيضاً الكائن.

object retention ــ حجز الكائن. الإجراء الذي يعتمده برنامج إدارة الكائنات لإبقاء كائن في الذاكرة. وعند إزالة آخر مرجع إلى كائن، يحذف برنامج إدارة الكائنات كائن مؤقّت من الذاكرة. راجع أيضاً حجز الإسم.

object service ــ خدمة الكائن. خدمة نظام مرئيّة لنمط المستعمل لمناولة كائن. وفي Windows NT، تقوم خدمة كائن عادة بقراءة صفات كائن أو تغييرها وتستعمل مبدئياً من قبل الأنظمة الفرعيّة المحميّة.

object table ــ جدول الكائنات. بنية بيانات خاصة لمعالجة تحتوي مقابض إلى كل الكائنات التي فتحتها شعَب المعالجة. راجع أيضاً مقبض الكائن.

object type ــ نوع الكائن. نوع بيانات مجرّدة، مجموعة من الخدمات التي تعمل على حالات آنية لنوع البيانات ومجموعة من صفات الكائن. يعرّف نوع كائن بإستعمال كائن نوع. راجع أيضاً صفة الكائن وكائن النوع.

Open Systems Interconnection (OSI) reference model ــ النموذج المرجعي للتوصيل البيني للأنظمة المنفتحة. نموذج براجيّات معرّف من قبل المنظمة الدوليّة للمواصفات القياسية التي تصنّف قياسياً مستويات الخدمة وأنواع تفاعل الحواسيب المشبّكة. يُعرِّف النموذج المرجعي OSI سبع طبقات من إتصالات الحاسوب ومسؤوليّة كل طبقة.

OSI. التوصيل البيني للأنظمة المنفتحة.

page ــ صفحة. كتل من العناوين الظاهريّة المتجاورة التي ينسخها برنامج إدارة الذاكرة الظاهريّة (VM) من الذاكرة إلى القرص ومنه خلال عملية الترتيب في صفحات. راجع أيضاً إطار الصفحة والترتيب في صفحات.

paged pool ــ المجموعة المرتّبة في صفحات. قسم من ذاكرة النظام الذي يمكن ترتيبه في صفحات إلى قرص. قارن مع المجموعة غير المرتّبة في صفحات.

page Fault ــ خطأ صفحة. مصيدة معالج تحصل عندما تشير شعبة تنفيذ إلى عنوان ظاهري موجود على صفحة غير صالحة. راجع أيضاً الصفحة غير الصالحة والترتيب في صفحات.

page frame ــ إطار الصفحة. كتلة من العناوين الفعليّة المتجاورة المستعملة لتخزين محتويات صفحة ظاهريّة. يحدّد حجم إطار الصفحة (وغالباً حجم الصفحة) من قبَل المعالج. وعلى معظم الأنظمة، يكون حجم الصفحة وحجم إطار الصفحة هو نفسه. راجع أيضاً الصفحة والترتيب في صفحات.

page frame database ـ **قاعدة بيانات إطار الصفحة**. بنية بيانات يستعملها برنامج إدارة الذاكرة الظاهريّة (VM) لتسجيل حالة كل أُطُر الصفحات الفعليّة. راجع أيضاً إطار الصفحة.

pager ـ **ناقل الصفحة**. مكوّن برنامج إدارة الذاكرة الظاهريّة (VM) الذي ينفّذ عملية الترتيب في صفحات. راجع أيضاً الترتيب في صفحات.

page table ـ **جدول الصفحات**. جدول خاص لمعالجة يستعمله برنامج إدارة الذاكرة الظاهريّة (VM) لتخطيط العناوين الظاهريّة إلى عناوين الذاكرة الفعليّة أو إلى مواقع قرص. يتألف جدول الصفحات من إدخالات جدول الصفحات (PTEs). راجع أيضاً إدخال جدول الصفحات والترتيب في صفحات.

(PTE) page table entry ـ **إدخال جدول الصفحات**. إدخال في جدول صفحات معالجة. وهو يحتوي المعلومات الضروريّة لنظام الذاكرة الظاهريّة لتحديد موقع صفحة عندما تستعمل شعبة عنوان غير صالح. يعتمد حجم ونسق إدخالات PTE على المعالج. راجع أيضاً الصفحة غير الصالحة وجدول الصفحات.

paging ـ **الترتيب في صفحات**. عملية ذاكرة ظاهريّة حيث تنقل براجيّات إدارة الذاكرة الصفحات من الذاكرة إلى القرص عندما تمتلىء الذاكرة الفعليّة. وعندما تتمكّن الشعبة من الوصول إلى صفحة غير موجودة في الذاكرة، يحصل خطأ صفحة ويستعمل برنامج إدارة الذاكرة جداول الصفحات لتحديد موقع الصفحة المطلوبة على القرص وتحميلها إلى الذاكرة. راجع أيضاً الصفحة غير الصالحة وخطأ الصفحة وجدول الصفحات.

paging file ـ **ملفّ الترتيب في صفحات**. ملفّ نظام يحتوي محتويات صفحات ظاهريّة تمّ إخراجها من الذاكرة. راجع أيضاً المخزن المساند والملفّ المخطّط.

parent process ـ **المعالجة الأم**. معالجة أنشأت معالجة أخرى، تسمى معالجة تابع. تتأصّل معالجة التابع بعض أو كل موارد المعالجة الأم. قارن مع المعالجة التابع.

placement policy ـ **سياسة الوضع**. خوارزميّة يستعملها نظام ذاكرة ظاهريّة ليقرّر مكان وضع البيانات التي يرتّبها في صفحات من قرص في الذاكرة الفعليّة. يستعمل برنامج إدارة الذاكرة الظاهريّة NT (VM) سلسلة من قائمات الصفحات الداخلة أولاً الخارجة أولاً (FIFO) لتعقّب الصفحات الخالية ولإسترداد صفحة خالية عند تحميل المعلومات من القرص بعد حصول خطأ صفحة.

port ـ منفّذ. قناة إتصال تتّصل بواسطتها معالجة مستضاف مع نظام فرعي محمي. تستخدم المنافذ ككائنات Windows NT. راجع أيضاً إستدعاء الإجراء المحلي.

power notify object ـ كائن الإبلاغ عن الطاقة. كائن نواة يتيح لمسيّقات الجهاز تسجيل روتين إستعادة الطاقة مع النواة. وهو كائن تحكّم يحتوي مؤشّراً إلى روتين مسيّق جهاز تستدعيه النواة عندما ترجع الطاقة بعد إنقطاعها.

power status object ـ كائن حالة الطاقة. كائن نواة يتيح لمسيّقات الجهاز لتحديد إنقطاع الطاقة. وهو كائن تحكّم يحتوي متغيّر Boolean يستطيع أن يختبره مسيّق جهاز قبل متابعة عمليّة مقاطعة. وإذا إنقطعت الطاقة، لا يبدأ المسيّق التشغيل.

PPTE. إدخال جدول صفحة نموذج أولي.

preempt ـ التملّك بالشفعة. لمقاطعة تنفيذ شعبة عندما تصبح شعبة بأولويّة أعلى جاهزة للتنفيذ وللتبديل السياقي إلى شعبة بأولويّة أعلى. راجع المهام المتعدّدة الشفعيّة.

preemptive multitasking ـ المهام المتعدّدة الشفعيّة. نسق مهام متعدّدة حيث يقاطع نظام التشغيل دورياً تنفيذ شعبة وينفّذ الشعَب المنتظرة الأخرى. يمنع التملّك بالشفعة الشعبة من إحتكار المعالج ويتيح تشغيل شعبة أخرى. راجع أيضاً حصة الوقت.

primary domain ـ الميدان الأولي. ميدان الشبكة يربط به حساب مستعمل معيّن. راجع أيضاً ميدان الشبكة.

privileged processor mode ـ نمط المعالج بأفضليّة. راجع أيضاً نمط النواة.

process ـ معالجة. قسم منطقي من العمل في نظام تشغيل. وفي Windows NT، فهو يتألف من فسحة عنوان ظاهري وبرنامج قابل للتنفيذ وشعبة تنفيذ واحدة أو أكثر وبعض الأجزاء من حصص موارد المستعمل وموارد النظام التي حدّدها نظام التشغيل لشعَب المعالجة. وهي تستخدم ككائن. راجع أيضاً الشعبة.

process context ـ سياق المعالجة. راجع سياق الشعبة.

process affinity ـ صلة المعالجة. مجموعة المعالجات حيث تستطيع الشعبة الإشتغال.

process tree ـ شجرة المعالجة. تسلسل هرمي لمعالجات الأم والتابع المحافظ عليها من قبَل النظامين الفرعيين POSIX و OS/2.

protected subsystem ـ النظام الفرعي المحمي. معالجة ملقّم تنفّذ وظائف نظام التشغيل.

يعمل كل نظام فرعي محمي في Windows NT في نمط المستعمل مع فسحة عنوان خاصة. راجع أيضاً النظام الفرعي للمحيط والنظام الفرعي المتكامل.

protocol ـ بروتوكول. مجموعة من القواعد والمصطلحات التي يعتمدها حاسوبان لتمرير الرسائل عبر وسيط شبكة. تستخدم براجيّات تشبيك الحواسيب عموماً مستويات متعدّدة من البروتوكولات المرتّبة في طبقات فوق بعضها البعض.

protocol stack ـ تكديس البروتوكولات. مجموعة وتتابع بروتوكولات الشبكة المستعملة لإرسال طلب شبكة من ماكنة واحدة إلى أخرى. راجع أيضاً البروتوكول.

(PPTE) prototype page table entry ـ إدخال جدول صفحات نموذج أولي. بنية بيانات تبدو مشابهة لإدخال جدول صفحات عادي (PTE) لكنه يشير إلى إطار صفحة مشارك من قبل أكثر من معالجة واحدة. راجع أيضاً إدخال جدول الصفحات وكائن القسم.

provider ـ الموفّر. إسم عام لبراجيّات تجعل من Windows NT كمستضاف لملقّم شبكة عن بُعد.

provider interface ـ تداخل الموفّر. تداخل برمجة يتيح لبائعي الشبكات توفير أنظمة الملفّات عن بُعد للتصفّح من قبل التطبيقات بإستعمال تداخل البرمجة التطبيقيّة (API) في Windows NT. راجع أيضاً مسلّك الموفّر المتعدّد.

PTE. إدخال جدول صفحات.

quick LPC ـ إستدعاء إجراء محلي سريع. شكل إستدعاء إجراء محلي (LPC) مستعمل من قبل أجزاء من النظام الفرعي Win 32 ومستضافاته. يزيد إستدعاء LPC السريع سرعة تمرير الرسالة بواسطة تجاوز كائنات المنفّذ وتخزين الرسائل في ذاكرة مشاركة واستعمال آلية مزامنة داخليّة. راجع أيضاً إستدعاء الإجراء المحلي والمنفّذ.

quota ـ الحصة. حدود موارد مطبقة على حسابات المستعمل. يغرّم برنامج إدارة الكائنات معالجة بضعة أجزاء من حصة المستعمل في كل مرة تنشىء فيها إحدى شعَب المعالجة أو تفتح مقبضاً إلى كائن. وعند إستنفاذ الحصة، لن تتمكّن معالجات المستعمل من إنشاء كائنات أو فتح مقابض كائن إلى أن تفلت المعالجات بعض الموارد.

raise an exception ـ إنشاء إستثناء. لنقل التحكّم بتعمّد إلى مناول إستثناء عند حصول إستثناء. تنشىء البراجيّات الإستثناء عند حصول أخطاء أو ظروف غير متوقّعة. راجع أيضاً الإستثناء ومناول الإستثناء.

redirector ـ **مغيِّر الوجهة**. براجيّات تشبيك الحواسيب تقبل طلبات الدخل / الخرج للملفّات عن بُعد وأنابيب مسمّاة أو شقوقاً بريديّة وترسلها («تغيّر وجهتها») إلى ملقّم شبكة على ماكنة أخرى. تستخدم مغيِّرات الوجهة كمسيِّقات لنظام ملفّات في Windows NT. راجع أيضاً ملقّم الشبكة.

(RISC) reduced instruction set computer ـ **حاسوب مجموعة التعليمات المخفّضة**. معالج يستخدم عدداً صغيراً من التعليمات البسيطة المستعملة في ترابط لتنفيذ عمليات أكثر قوة. وبسبب سهولة التعليمات واستعمالها لعدد كبير من المسجّلات، يستغرق كل منها دورة ساعة واحدة للتنفيذ، ويستطيع المعالج التشغيل عند سرعات ساعة أعلى مقارنة مع حواسيب مجموعة التعليمات المعقّدة (CISC) قارن مع حاسوب مجموعة التعليمات المعقّدة.

(RPC) remote procedure call ـ **إستدعاء الإجراء عن بُعد**. وسيلة تمرير رسائل تتيح لتطبيق موزّع إستدعاء الخدمات المتوفّرة على الماكنات المختلفة في شبكة دون إعتبار لموقعها. ويتمّ تناول عمليات الشبكة عن بُعد تلقائياً. يوفّر إستدعاء RPC معاينة إجرائيّة بدلاً من معاينة أساسها النقل لعمليات مشبّكة. قارن مع إستدعاء الإجراء المحلي.

replacement policy ـ **سياسة الإستبدال**. الخوارزميّة المستعملة من قبل نظام الذاكرة الظاهريّة لتحديد الصفحة الظاهريّة الواجب إزالتها من الذاكرة لتوفير مكان للبيانات التي يتمّ ترتيبها في صفحات من القرص. يستعمل النظام Windows NT سياسة إستبدال محلي دنيا مستعملة مؤخراً. راجع أيضاً سياسة الإستبدال المحلي.

reserved memory ـ **الذاكرة المحجوزة**. مجموعة من عناوين الذاكرة الظاهريّة التي حدّدتها شعبة. راجع أيضاً الذاكرة المعتمدة.

robustness ـ **القوّة**. قدرة برنامج على العمل بشكل جيّد أو مواصلة العمل بشكل جيّد في حالات غير متوقّعة.

RPC ـ إستدعاء الإجراء عن بُعد.

RPC transport provider interface ـ **تداخل موفّر النقل لإستدعاء إجراء عن بُعد**. مكتبة DLL تعمل كتداخل بين وسيلة إستدعاء إجراء عن بُعد (RPC) وبراجيّات نقل الشبكة. وهو يتيح إرسال إستدعاءات RPC على بروتوكولات نقل متنوّعة.

SAM database ـ **قاعدة بيانات برنامج إدارة حسابات الأمان**. قاعدة بيانات معلومات الأمان

التي تشمل أسماء حسابات المستعمل وكلمات السرّ. ويتمّ إدارتها من قبل Windows User Manager. راجع أيضاً برنامج إدارة حسابات الأمان.

script ـ النص المكتوب. نظام أحرف مستعمل للكتابة في لغة واحدة أو أكثر.

section object ـ كائن القسم. كائن يمثّل الذاكرة المشاركة جهدياً من قبل معالجتين أو أكثر. تستطيع معالجة أيضاً إنشاء كائن قسم غير مسمّى يمثّل ذاكرة خاصة. راجع أيضاً المعاينة.

secure logon facility ـ وسيلة التسجيل الآمن. براجيّات في نظام تشغيل آمن يراقب فئة معيّنة من أجهزة التسجيل لضمان إدخال كل المستعملين التعريف الصالح قبل الإتاحة لهم بالوصول إلى النظام.

security account manager (SAM) ـ برنامج إدارة حساب الأمان. نظام فرعي محمي في Windows NT يحافظ على قاعدة بيانات SAM ويوفّر تداخل برمجة تطبيقيّة (API) للوصول إلى قاعدة البيانات. راجع أيضاً قاعدة بيانات SAM.

security descriptor ـ واصف الأمان. بنية بيانات ملحقة بكائن تحمي الكائن من الوصول غير المسموح به. وهي تحتوي قائمة التحكُّم بالوصول (ACL) وتتحكّم بالتدقيق على الكائن. راجع أيضاً قائمة التحكُّم بالوصول والتدقيق.

security ID ـ بطاقة تعريف الأمان. إسم فريد لجهة الوقت والفسحة. يعرّف مستعمل مسجّل إلى نظام الأمان. تستطيع بطاقات ID الأمان (SIDs) تعريف إما مستعمل إفرادي أو مجموعة من المستعملين. توافق عادة بطاقة ID أمان مع معرّف تسجيل المستعمل.

security reference monitor ـ مراقب مراجع الأمان. مكوّن البرنامج التنفيذي NT الذي يقارن صفة الوصول لمعالجة مع قائمة التحكُّم بالوصول (ACL) لكائن لتحديد الإتاحة لشعَب المعالجة من فتح مقبض إلى الكائن.

security subsystem ـ النظام الفرعي للأمان. نظام فرعي متكامل يسجّل سياسات الأمان الفاعلة لحاسوب محلي وتشارك في تسجيل المستعملين. راجع أيضاً النظام الفرعي المتكامل.

server ـ الملقّم. معالجة بشعبة واحدة أو أكثر تقبل الطلبات من معالجات المستضاف. وهي تستخدم مجموعة من الخدمات المتوفّرة للمستضافات الشغّالة إما على نفس الحاسوب أو على حواسيب متنوّعة في شبكة موزّعة. راجع أيضاً المستضاف وإستدعاء الإجراء المحلي وملقّم الشبكة وإستدعاء الإجراء عن بُعد.

server message block (SMB) protoc ــ بروتوكول كتلة رسائل الملقّم. بروتوكول شبكة مستعمل أصلاً في شبكات Microsoft Networks واعتمدت لاحقاً في براجيّات تشبيك الحواسيب. وهو يعرّف نسقاً معيّناً لحزمات البيانات الواجب إرسالها عبر الشبكة. يستعمل مغيِّر الوجهة والملقّم الداخلي في Windows NT بروتوكولات SMB للإتصال مع بعضها البعض ومع الحواسيب على شبكات LAN Manager، راجع أيضاً ملقّم الشبكة والبروتوكول ومغيِّر الوجهة.

server service ــ خدمة الملقّم. خدمة شبكة تزوّد تداخل برمجة تطبيقيّة (API) لتطبيق في نمط المستعمل وذلك لإدارة ملقّم الشبكة في Windows NT. راجع أيضاً الملقّم.

service ــ الخدمة. معالجة ملقّم تنفّذ وظيفة نظام معيّنة وتوفّر غالباً تداخل برمجة تطبيقيّة (API) ليتمّ إستدعاءها من قبل المعالجات الأخرى. تمكّن خدمات Windows NT بواسطة الإستدعاء RPC وهذا يعني أنه يمكن إستدعاء روتينات API العائدة لها من ماكنات عن بُعد.

service controller ــ وحدة التحكُّم بالخدمة. مكوّن تشبيك الحواسيب الذي يحمّل ويبدأ خدمات النظام Windows NT. وهو يحمّل أيضاً العديد من مسيّقات Windows NT ويلغي تحميلها، بما في ذلك مسيّقات الجهاز ومسيّقات نقل الشبكة. راجع أيضاً الخدمة.

SID ــ بطاقة ID للأمان.

signaled state ــ الحالة المؤشّرة. صفة كل كائن ذات نوع يدعم المزامنة، وعندما تضبط النواة الكائن إلى الحالة المؤشّرة، يتمّ إفلات الشعَب التي تنتظر الكائن من حالة الإنتظار (وفقاً لمجموعة من القواعد) وتصبح صالحة للتنفيذ. راجع أيضاً كائن الموزّع والحالة غير المؤشّرة والمزامنة.

single-byte coding scheme ــ مخطّط التشفير الأحادي البايت. مخطّط تشفير أحرف (مجموعة شيفرات) مثل Windows ANSI الذي يستعمل ثماني بتات لتمثيل كل حرف. راجع أيضاً الشيفرة الأحاديّة.

SMB ــ كتلة رسائل الملقّم.

SMP ــ المعالجة المتعدّدة المتناظرة.

spin lock ــ القفل الدوامي. آلية مزامنة مستعملة من قبل النواة وأجزاء البرنامج التنفيذي التي تضمن الوصول الحصري التبادلي إلى بنية بيانات النظام العامة عبر معالجات متعدّدة.

تؤخّر شعبة تنتظر الحصول على قفل دوامي المعالج إلى أن تحصل على القفل الدوامي. راجع أيضاً المنع المتبادل.

STREAMS ـــ محيط تطوير مسيّق يزوّده النظام Windows NT لإنشاء أو إنفاذ مسيّقات نقل الشبكة.

structured exception handling ـــ المناولة الإستثنائيّة البنيويّة. طريقة إلتقاط الظروف غير المتوقّعة والإجابة عليها بثبات في كل نظام التشغيل. يصدر نظام التشغيل (أو العتاد) الإستثناء عند حصول حدث نظام غير عادي وتنقل النواة تلقائياً التحكّم إلى مناول إستثناء. راجع أيضاً الإستثناء ومناول الإستثناءات.

stub procedure ـــ الإجراء الجذلي. إجراء في مكتبة الربط الدينامي (DLL) يستخدم كنقطة إدخال لتداخل برمجة تطبيقيّة (API). وعندما يستدعي تطبيق مستضاف روتين API، ينظّم الإجراء الجذلي بارامترات API التي يستلمها إلى رسالة ويرسلها إما إلى ملقّم محلي (نظام فرعي) أو إلى ملقّم عن بُعد على الشبكة. راجع أيضاً إستدعاء الإجراء المحلي والمنظّم وإستدعاء الإجراء عن بُعد.

symbolic link object ـــ كائن الربط الرمزي. كائن برنامج تنفيذي NT يترجم إسم كائن واحد إلى آخر.

symmetric multiprocessing (SMP) ـــ المعالجة المتعدّدة المتناظرة. نظام تشغيل معالجة متعدّدة يتيح تشغيل شيفرة نظام التشغيل على أي معالج خالي في حاسوب متعدّد المعالجات. توفّر عادة أنظمة المعالجة المتعدّدة المتناظرة نتائج أفضل وتوفّريّة أكبر مما توفّره الأنظمة المتعدّدة المعالجة غير المتناظرة. قارن مع المعالجة المتعدّدة غير المتناظرة وراجع أيضاً المعالجة المتعدّدة.

synchronization ـــ المزامنة. قدرة شعبة واحدة على التوقّف خلال التنفيذ وإنتظار قناة شعبة أخرى بتنفيذ عمليّة. وفي Windows NT، تنتظر شعبة ضبط شعبة أخرى لكائن مزامنة إلى الحالة المؤشّرة. راجع أيضاً الحالة المؤشّرة وكائنات المزامنة.

synchronization objects ـــ كائنات المزامنة. مجموعة من كائنات البرنامج التنفيذي NT المرئيّة في نمط المستعمل ذات أنواع كائنات تدعم المزامنة. وهي تشمل الشعَب والمعالجات والأحداث وزوج الأحداث والإعلام الإشاري والمؤقّتات والخوافت والملفّات. تستطيع شعبة إنتظار ضبط كائن مزامنة إلى الحالة المؤشّرة من قبل شعبة أخرى. يحتوي كل كائن مزامنة على كائن موزّع نواة فيه. راجع أيضاً كائن الموزّع والحالة المؤشّرة والمزامنة.

synchronous ــ المتزامن. يحصل عند وقت معيّن كنتيجة مباشرة لتنفيذ تعليمة ماكنة معيّنة. قارن مع عدم التزامن.

synchronous I/O ــ الدخل / الخرج المتزامن. نموذج دخل / خرج حيث يصدر تطبيق طلب دخل / خرج ولا يرجع نظام الدخل / الخرج التحكّم إلى التطبيق إلى أن يتمّ طلب الدخل / الخرج. قارن مع الدخل / الخرج غير المتزامن.

TCP/IP transport ــ بروتوكول النقل لبروتوكول التحكّم بالإرسال / بروتوكول **Internet**. بروتوكول نقل شبكة مناطقيّة واسعة أوليّة في Windows NT. وهو يتيح للنظام Windows NT الإتصال مع الأنظمة على شبكات TCP/IP والمشاركة في لوحة الإعلانات الشهريّة التي تعتمد على النظام UNIX والأخبار والخدمات البريديّة الإلكترونيّة.

TDI ــ تداخل مسيّق النقل.

termination handler ــ مناول الإنهاء. مناول إستثناء يتيح لتطبيق ضمان تنفيذ كتلة معيّنة من الشيفرات دائماً، حتى إذا انتهت الشيفرة بطريقة غير متوقّعة. تحتوي غالباً مناولات الإنهاء شيفرة تخلّي الموارد المحدّدة بحيث في حال إنتهى إجراء بطريقة غير متوقّعة، تفلت الموارد مجدّداً إلى النظام. راجع أيضاً مناول الإستثناءات.

thread ــ الشعبة. وحدة مستقلّة قابلة للتنفيذ تعود إلى معالجة واحدة (وفقط واحدة). وهي تتألف من عدّاد برنامج وتكديس في نمط المستعمل وتكديس في نمط النواة ومجموعة من قيم المسجّل. وتحتوي كل الشعَب من معالجة على وصول متكافىء إلى فسحة عنوان المعالجة ومقابض كائن وموارد أخرى. وتستخدم الشعَب ككائنات. راجع أيضاً المعالجة.

thread context ــ سياق الشعبة. البيانات المتطايرة المتعلّقة بتنفيذ شعبة. وهو يشمل محتويات مسجّلات النظام وفسحة العنوان الظاهري العائد إلى معالجة الشعبة. راجع أيضاً التبديل السياقي.

thread dispatching ــ توزيع الشعبة. راجع التبديل السياقي.

thread object ــ كائن الشعبة. إستخدام شعبة في Windows NT. راجع أيضاً الشعبة.

thread of execution ــ شعبة تنفيذ. راجع الشعبة.

thread scheduling ــ جدولة الشعَب. عمليّة التدقيق في صفيفة الشعَب الجاهزة للتنفيذ وإنتقاء إحداها للتشغيل تالياً. تنفّذ هذه المهمة من قبل منظومة موزّع النواة NT. راجع أيضاً الموزّع.

tightly coupled system ـ النظام المقرون بأحكام. حاسوب متعدّد المعالجات حيث تشارك كل المعالجات الذاكرة العامة.

time quantum ـ الحصّة الزمنيّة. كميّة وقت مضبّطة مسبقاً حيث تتيح نواة نظام التشغيل تنفيذ شعبة قبل شفعها. راجع أيضاً التملُّك بالشفعة.

TLB ـ المخزن المؤقّت الجانبي للترجمة.

token object ـ كائن الصفة. راجع صفة الوصول.

topology ـ الطبولوجيا. التشكيل الفعلي للماكنات في شبكة.

translation lookaside buffer (TLB) ـ المخزن المؤقّت الجانبي للترجمة. صفيفة ذاكرة تحتوي تخطيطات العنوان الظاهري ـ إلى ـ الفعلي للصفحات التي إستُعملت مؤخراً في كل النظام. وتحتوي كل من معالجات MIPS ومعالجات Intel على مخازن TLB لكن بنيتها وعمليّاتها تعتمد على العتاد.

transport driver interface (TDI) ـ تداخل مسيّق النقل. تداخل Windows NT لمغيّرات وجهة الشبكة والملقّمات لتستعمل في إرسال الطلبات المتعلّقة بالشبكة إلى مسيّقات النقل. يوفّر التداخل إستقلاليّة النقل لهذه المكوّنات عن طريق تجريد المعلومات الخاصة بالنقل.

trap ـ المصيدة. آلية معالج لإلتقاط شعبة تنفيذ عند حصول حدث غير إعتيادي (مثل الإستثناء أو المقاطعة) ونقل التحكّم إلى موقع ثابت في الذاكرة. راجع أيضاً مناول المصيدة.

trap frame ـ إطار المصيدة. بنية بيانات ينشئها مناول مصيدة النواة عند حصول مقاطعة أو إستثناء. وهو يسجّل حالة المعالج الذي يتيح للنواة مواصلة تنفيذ الشعبة المقاطعة بعد مناولة الحالة. راجع أيضاً مناول المصيدة.

trap handler ـ مناول المصيدة. جسم شيفرة تحفزه العتاد عند حصول مقاطعة إستثناء. وهو يحدّد نوع الظرف الذي حصل وينقل التحكُّم إلى روتين مناولة. راجع أيضاً المصيدة.

trusted domain relationship ـ علاقة الميدان الموثوقة. علاقة ثقة موجودة بين ميداني شبكة. راجع أيضاً ميدان الشبكة وعلاقة الثقة.

trust relationship ـ علاقة الثقة. بند أمان يعني أن محطة عمل واحدة أو ملقّم شبكة تثق بوحدة تحكُّم بميدان للتحقّق من أصالة تسجيل مستعمل نيابة عنه. وتستطيع وحدة تحكُّم

واحدة بميدان أن تثق بوحدة تحكّم بميدان في ميدان آخر للتحقّق من أصالة تسجيل. راجع أيضاً وحدة التحكّم بالميدان.

type object ــ كائن النوع. كائن نظام داخلي يعرّف الصفات العامة لفئة كائنات. وتحتوي كل حالة آنية لكائن على مؤشّر إلى كائن نوع موافق. راجع أيضاً نوع الكائن.

UNC ــ مصطلح التسمية المتناسق.

Unicode ــ الشيفرة الآحاديّة. مواصفات تشفير بأحرف من 16 بت بعرض ثابت تستطيع تمثيل كل نصوص العالم المكتوبة. راجع أيضاً النص المكتوب.

uniform naming convention (UNC) names ــ أسماء مصطلح التسمية المتناسق. أسماء الملفّات أو أسماء الموارد الأخرى التي تبدأ بالنضيد \، حيث تشير إلى وجودها على ماكنة عن بُعد.

(UPS) uninterruptible power supply ــ مصدر طاقة لا ينقطع. منظومة بطّاريّة مساندة ملحقة بالحاسوب يتيح إبقاء محتويات الذاكرة كما هي لفترة تكفي لكي ينفّذ نظام التشغيل توقيف منظّم للنظام في حال إنقطاع الطاقة الكهربائيّة.

UPS ــ مصدر طاقة لا ينقطع.

user mode ــ نمط المستعمل. نمط المعالج من دون أفضليّة الذي تشتغل فيه شيفرة التطبيق. وتستطيع شعبة تشتغل في نمط المستعمل الوصول إلى النظام فقط عن طريق إستدعاء خدمات النظام. قارن مع نمط النواة.

valid page ــ صفحة صالحة. صفحة ظاهريّة موجودة في الذاكرة الفعليّة ومتوفّرة فوراً. راجع أيضاً الصفحة غير الصالحة والصفحة.

VDM ــ ماكنة DOS الظاهريّة.

view ــ المعاينة. جزء من كائن قسم تخطّطه المعالجة في فسحة العنوان الظاهري. وتستطيع المعالجة تخطيط معاينات متعدّدة وحتى متراكبة لقسم. راجع أيضاً الخريطة وكائن القسم.

virtual address space ــ فسحة العنوان الظاهري. مجموعة العناوين المتوفّرة لتستعمل من قبل شعبة معالجة. وفي Windows NT، تحتوي كل معالجة على فسحة عنوان ظاهري فريد من 2^{32} بايت (4 جيغا بايت). راجع أيضاً الذاكرة الظاهريّة.

virtual circuit ــ الدائرة الظاهريّة. قناة إتصال ظاهريّة بين ماكنتين. وتتكثّف دورات عمل الشبكة المتعدّدة عبر دائرة ظاهريّة واحدة.

virtual DOS machine (VDM) ــ ماكنة DOS ظاهريّة. نظام فرعي محمي يزوّد محيط MS-DOS كاملاً وكونسول حيث يشتغل تطبيق MS-DOS. ويستطيع أي عدد من ماكنة VDM الإشتغال في نفس الوقت. راجع أيضاً الكونسول.

virtual file ــ الملفّ الظاهري. أي مصدر أو مقصد لدخل / خرج قابل للوصول مثل الملفّ. وفي البرنامج التنفيذي NT، تنفّذ كل طلبات الدخل / الخرج على الملفّات الظاهريّة الممثّلة من قبَل كائنات الملفّ والتي يتمّ الوصول إليها بإستعمال مقابض الملفّ. راجع أيضاً كائن الملفّ.

virtual memory (VM) ــ الذاكرة الظاهريّة. معاينة محليّة للذاكرة لا تتوافق بالضرورة مع البنية الفعليّة للذاكرة. راجع أيضاً إدارة الذاكرة الظاهريّة.

virtual memory management ــ إدارة الذاكرة الظاهريّة. نظام إدارة ذاكرة يوفّر فسحة عنوان كبيرة لكل معالجة عن طريق تخطيط العناوين الظاهريّة للمعالجة على عناوين فعليّة عندما تستعمل من قبل شعَب المعالجة. وعندما تمتلىء الذاكرة الفعليّة، فإنها تبادل محتويات الذاكرة المحدّدة إلى القرص وتعيد تحميلها من القرص عند الطلب. تتيح إدارة الذاكرة الظاهريّة للمبرمجين إنشاء البرامج وتشغيلها والتي تستعمل ذاكرة أكثر مما هو متوفّر فعلياً على الحواسيب. ولأن وضع البيانات في الذاكرة محكوم من قبل نظام الذاكرة الظاهريّة، يمكن فصل كل فسحة عنوان معالجة وحمايتها من الأخرى. راجع أيضاً الخريطة والترتيب في صفحات.

virtual memory (VM) manager ــ برنامج إدارة الذاكرة الظاهريّة. مكوّن البرنامج التنفيذي NT الذي يستخدم الذاكرة الظاهريّة.

VM ــ ذاكرة ظاهريّة.

Win32 API ــ تداخل البرمجة التطبيقيّة في Win32. تداخل برمجة تطبيقيّة من 32 بت للأنظمة Windows/Ms-DOS و Windows NT. وهو يحدّث الإصدارات السابقة لتداخل البرمجة التطبيقيّة (API) في Windows مع قدرات نظام التشغيل المعقّدة والأمان وروتينات API لعرض التطبيقات النصّيّة في إطار.

Windows NT ــ نظام التشغيل Windows متطوّر في عائلة أنظمة التشغيل Windows. وسويّة مع Pen Windows و 16-bit Windows، يتيح هذا النظام تشغيل تطبيقات Windows على حواسيب بدءاً من المفكرات الصغيرة إلى محطات العمل المتعدّدة المعالجات الكبيرة وماكنات الملقّم. كذلك يشتغل النظام Windows NT تطبيقات MS-DOS و POSIX

و OS/2 عن طريق إستخدام الملقّمات في نمط المستعمل التي تسمى الأنظمة الفرعيّة المحميّة. راجع أيضاً النظام الفرعي المحمي.

Windows on Win32 (WOW) ــ نظام فرعي محمي يشتغل ضمن معالجة ماكنة DOS ظاهريّة (VDM). وهو يوفّر محيط 16-bit Windows قادر على تشغيل أي عدد من تطبيقات 16-bit Windows على النظام Windows NT.

working set ــ مجموعة العمل. مجموعة الصفحات الظاهريّة الموجودة في الذاكرة الفعليّة في أية لحظة لمعالجة معيّنة.

workstation service ــ خدمة محطة العمل. خدمة شبكة تزوّد روتينات تداخل البرمجة التطبيقيّة (API) لتطبيق في نمط المستعمل وذلك لإدارة مغيِّر الوجهة في Windows NT. راجع أيضاً الخدمة.

WOW ــ Windows على Win32.

الدار العربية للعلوم
Arab Scientific Publishers

Microsoft
PRESS

INSIDE WINDOWS NT™

HELEN CUSTER

FOREWORD BY DAVID N. CUTLER